J 6

```
MMM         MMM  TTTTTTTTTTTTTTT     AAAAAAAAA        AAAAAAAAA        CCCCCCCCCCCC  PPPPPPPPPPPP
MMM         MMM  TTTTTTTTTTTTTTT     AAAAAAAAA        AAAAAAAAA        CCCCCCCCCCCC  PPPPPPPPPPPP
MMM         MMM  TTTTTTTTTTTTTTT     AAAAAAAAA        AAAAAAAAA        CCCCCCCCCCCC  PPPPPPPPPPPP
MMMMMM   MMMMMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMMMMM   MMMMMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMMMMM   MMMMMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMM   MMM   MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMM   MMM   MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMM   MMM   MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPPPPPPPPPPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPPPPPPPPPPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPPPPPPPPPPP
MMM         MMM        TTT        AAAAAAAAAAAAAAA  AAAAAAAAAAAAAAA  CCC              PPP
MMM         MMM        TTT        AAAAAAAAAAAAAAA  AAAAAAAAAAAAAAA  CCC              PPP
MMM         MMM        TTT        AAAAAAAAAAAAAAA  AAAAAAAAAAAAAAA  CCC              PPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA     CCCCCCCCCCCC  PPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA     CCCCCCCCCCCC  PPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA     CCCCCCCCCCCC  PPP
```

J 9

```
**FILE**ID**MOUVOL




MM      MM    OOOOOO    UU      UU  VV      VV    OOOOOO    LL
MM      MM    OOOOOO    UU      UU  VV      VV    OOOOOO    LL
MMMM  MMMM  OO      OO  UU      UU  VV      VV  OO      OO  LL
MMMM  MMMM  OO      OO  UU      UU  VV      VV  OO      OO  LL
MM  MM  MM  OO      OO  UU      UU  VV      VV  OO      OO  LL
MM  MM  MM  OO      OO  UU      UU  VV      VV  OO      OO  LL
MM      MM  OO      OO  UU      UU  VV      VV  OO      OO  LL
MM      MM  OO      OO  UU      UU  VV      VV  OO      OO  LL
MM      MM  OO      OO  UU      UU  VV      VV  OO      OO  LL
MM      MM  OO      OO  UU      UU  VV      VV  OO      OO  LL
MM      MM  OO      OO  UU      UU    VV  VV    OO      OO  LL              ....
MM      MM  OO      OO  UU      UU    VV  VV    OO      OO  LL              ....
MM      MM    OOOOOO    UUUUUUUUUU      VV        OOOOOO    LLLLLLLLLL      ....
MM      MM    OOOOOO    UUUUUUUUUU      VV        OOOOOO    LLLLLLLLLL      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

MOU
VO

```
0
0 MODULE MOUVOL (LANGUAGE (BLISS32) ,
0               IDENT = 'V04-000'
0               ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:  MTAACP
1 !
1 ! ABSTRACT:
1 !       This module mounts a volume
1 !
1 !
1 ! ENVIRONMENT:
1 !
1 !       VMS operating system, including privileged system services
1 !       and internal exec routines.
1 !
1 !--
1
1 !
1 !
1 ! AUTHOR:  D. H. GILLESPIE,       CREATION DATE:  24-AUG-1977
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-013 HH0041          Hai Huang               24-Jul-1984
1 !               Remove REQUIRE 'LIBD$:[VMSLIB.OBJ]MOUNTMSG.B32'.
1 !
1 !       V03-012 MMD0288         Meg Dumont,     10-Apr-1984  14:14
1 !               Fix to the return from $MTACCESS code where ACCESS could
1 !               be set to normal processing before all the error conditions
1 !               where checked.
```

MOU
V04

```
1 !
1 !     V03-011 LMP0221         L. Mark Pilant,         28-Mar-1984  14:45
1 !             Change UCB$L_OWNUIC to ORB$L_OWNER and UCB$W_VPROT to
1 !             ORB$W_PROT.
1 !
1 !     V03-010 MMD0271         Meg Dumont,     23-Mar-1984  9:34
1 !             Change the processing of the accessibility character fields
1 !             in the VOL1 label to call the installation
1 !             specific accessibility routine. The return from this
1 !             routine determines the users access to the volume.
1 !
1 !     V03-009 MMD0185         Meg Dumont,     6-Jul-1983  18:32
1 !             Make the default for AVL/AVR the same from the DCL call
1 !             and from the system service call.
1 !
1 !     V03-008 MMD0176         Meg Dumont,     26-May-1983  15:11
1 !             Fix to support new input to IOC$CVT_DEVNAM
1 !
1 !     V03-007 MMD0174         Meg Dumont,     9-May-1983  15:16
1 !             Fix to make IO_STATUS consistently defined within module
1 !
1 !     V03-006 MMD0164         Meg Dumont,     26-Apr-1983  9:43
1 !             Change the references to 80 to be the symbol ANSI_LBLSZ. Change
1 !             the reference to 240 to be the symbol SCRATCH_OFFSET.
1 !
1 !     V03-005 MMD0134         Meg Dumont,     12-Apr-1983  17:24
1 !             Added support for writng and interrupting the VOL1
1 !             OWNER IDENTIFIER field, so that it is no longer
1 !             treated as a VMS field, strictly. Bugfix to the AVL, AVR
1 !             code where MOUNT/INIT would not work under all circumstances.
1 !
1 !     V03-004 MMD0120         Meg Dumont,     29-Mar-1983  0:44
1 !             Added support for the VOL2 label inside the MTAACP
1 !
1 !     V03-003 MMD0103         Meg Dumont,     17-Feb-1983  13:14
1 !             Use GET_DEV_NAME to get the tape units device name. Added
1 !             the routine GET_DEV_NAME to call the system routine
1 !             IOC$CVT_DEVNAM to get the tape units name. Added the code
1 !             to do automatic volume recognition and labeling (AVR and
1 !             AVL).
1 !
1 !     V03-002 MMD0002         Meg Dumont,     3-Jan-1983  15:43
1 !             Allow user with read access to a tape to mount the tape
1 !             writelocked. Add modifier IO$M_CLRSEREXCP to all QIO's issued
1 !             by the MTAACP, necessary for the MSCP tape drives.
1
1 !
1 !     V03-001 MMD0001         Meg Dumont,     23-Mar-1982  10:16
1 !             Added a check for member UIC match when mounting a volume.
1 !
1 !     V02-014 DMW00071        David Michael Walp      21-Jan-1981
1 !             Handle Volume Invalid during verification
1
1 !     V02-014 DMW00059        David Michael Walp      7-Dec-1981
1 !             Moved Rename TRANSLATION_TABLE to ANSI_A_BAD, ANSI_A_GOOD
1 !
1 !     V02-013 DMW00036        David Michael Walp      17-Sep-1981
```

```
!               Correct error messages given ( MOUNT, REMOUNT switched )
!
!       V02-012 DMW00030        David Michael Walp      18-Aug-1981
!               Volume Access and ANSI 'a' character in Volume Names
!
!       V02-011 DMW00017        David Michael Walp      20-May-1981
!               Copy the new fields ( File-Set-Id and Vol_Acc ) then
!               creating New MVL.
!
!       V02-010 DMW00014        David Michael Walp      14-Mar-1981
!               Changed the calculation of the CCB address to GET_CCB
!
!       V02-009 REFORMAT        Maria del C. Nasr       30-Jun-1980
!
!       A0008   MCN0003         Maria del C. Nasr       15-Oct-1979   9:23
!               Add HDR3 processing
!
!**

LIBRARY 'SYS$LIBRARY:LIB.L32';

REQUIRE 'SRC$:MTADEF.B32';

REQUIRE 'LIBD$:[VMSLIB.OBJ]INITMSG.B32';

LINKAGE
    L$CHOOSE_UNIT   = JSB :  GLOBAL (CURRENT_VCB = 11)
                              NOTUSED (2, 3, 4, 5, 6, 7, 8, 9, 10),
    MVL_UCB         = CALL : GLOBAL (MVL_ENTRY = 9, UCB_LIST = 10,
                              CURRENT_VCB = 11);

FORWARD ROUTINE
    ASSUME_MOUNTED      : NOVALUE MVL_UCB,      ! assume correct volume is
                                                !  mounted
    CLPREV_MAKECUR      : NOVALUE MVL_UCB,      ! clear prev use of volume and
                                                !  make it current
    CHECK_ACCESS        : MVL_UCB,              ! check the access rights to a
                                                !  tape
    CHECK_RING          : COMMON_CALL,          ! check for the write ring
    CHOOSE_UNIT         : L$CHOOSE_UNIT,        ! choose unit for mount
    CLEAR_UNIT          : NOVALUE MVL_UCB,      ! clear last mounted volume
    CREATE_LABEL        : COMMON_CALL,          ! create a label for the volume
    GET_DEV_NAME        : COMMON_CALL NOVALUE,  ! given the UCB addr get dev nam
    MAKE_CUR_VOL        : NOVALUE MVL_UCB,      ! make volume current
    MAKE_VOL_ENTRY      : COMMON_CALL,          ! make volume entry in MVL
    OPERATOR_LBL        : MVL_UCB,              ! record operator supplied
                                                !  label
    SET_MVL_OVERIDE     : NOVALUE MVL_UCB;      ! set the MVL overide bit

EXTERNAL
    CURRENT_UCB         : REF BBLOCK,   ! address of current unit control block
    CURRENT_WCB         : REF BBLOCK,   ! address of current WCB
    IO_CHANNEL,                         ! acp io channel
    IO_STATUS           : VECTOR [2],   ! status of io
    MAIL_CHANNEL,
    WORK_AREA;
```

```
1 EXTERNAL ROUTINE
1       BLOCK,                                      ! block activity on current volume set
1       CHECK_PROT,                                         ! check VMS protection rights
1       ENABLE_MAIL_AST     : COMMON_CALL,
1       GET_CCB,                                    ! get the address of the CCB
1                                                   ! get devname given its UCB
1       GET_RECORD,                                 ! get the record tape is currently reading
1       IOC$CVT_DEVNAM      : L$IOC_CVT_DEVNAM ADDRESSING_MODE (ABSOLUTE),
1       PRINT_OPR_MSG       : L$PRINT_OPR_MSG,
1       PRINT_NOT_LABEL     : JSB,                  ! print not correct label
1       PROCESS_VOL2_LABEL,                         ! interpret the contents of VOL2
1       READ_BLOCK          : COMMON_CALL,          ! read mag tape block
1       REWIND_AND_WAIT     : COMMON_CALL,          ! rewind volume and wait for completion
1       SEND_ERRLOG,
1       SYS$FAO             : ADDRESSING_MODE (ABSOLUTE),     ! format ascii output
1       SYS$QIOW            : ADDRESSING_MODE (ABSOLUTE),     ! queue I/O and wait
1       SYS$SETIMR          : ADDRESSING_MODE (ABSOLUTE),     ! set time request
1       SYS$WAITFR          : ADDRESSING_MODE (ABSOLUTE),     ! wait for event
1       TAPE_OWN_PROT,                              ! determine the owner and
1                                                   ! protection of a tape
1       TERMINATE_VOL;                              ! terminate mount request
1
1 BIND
1       MAIL   = WORK_AREA  : BBLOCK [MSGSIZE],
1       MAILSZ = MAIL + MSGSIZE,
1       STARID = UPLIT ('DECFILE11A');
1
1 OWN
1       TAPE_OWNER,
1       TAPE_PROT           : BITVECTOR [16],
1       LABEL_SPEC          : BITVECTOR [1],        ! set if oper specified label
1       INFORM_OPER         : BITVECTOR [1],        ! set if oper should know that
1                                                   ! a mount happen without their
1                                                   ! involvment
1       CVT_DEVNAM          : VECTOR [MAX_DEVNAM_LENGTH,BYTE], ! Converted device
1                                                              ! name
1       CVT_DEVNAM_LENGTH   : BYTE;                 ! length of device name
```

```
GLOBAL ROUTINE MOUNT_VOL (VOL, FLAGS) : COMMON_CALL =

!++
!
! FUNCTIONAL DESCRIPTION:
!
!& REWRITE THIS DESCRIPTION
!&
!	This routine mounts the specified relative volume.  If it is
!	already mounted and the rewind flag is set then the volume will be
!	rewound and the VOL1 label rechecked.  If the volume is not mounted,
!	a request is issued to the operator, the process blocks until the oper
!	replies that the volume has been mounted.  Then if the label flag
!	is set, the VOL1 label is checked against the one entered at the
!	original mount time or one entered by the operator when indicating
!	that the volume was mounted.
!
! CALLING SEQUENCE:
!	MOUNT_VOL(ARG1,ARG2)
!
! INPUT PARAMETERS:
!	ARG1 - relative volume number to mount
!	ARG2 - flags
!		MOU$V_REWIND   - request rewind of volume
!		MOU$V_LBLCHECK - request check of label
!		MOU$V_CHKIFSPC - check the label only if the operator supplied
!		MOU$V_MOUNTERR - error on last mount, so force mount
!
! IMPLICIT INPUTS:
!	NONE
!
! OUTPUT PARAMETERS:
!	NONE
!
! IMPLICIT OUTPUTS:
!	volume is mounted and set current
!
! ROUTINE VALUE:
!	NONE
!
! SIDE EFFECTS:
!	NONE
!
!--

    BEGIN

    BIND
	SECONDS = UPLIT (-10000000, -1);	! one second in 100 nsec units

    MAP
	FLAGS	: BBLOCK;

    GLOBAL REGISTER
	MVL_ENTRY = 9	: REF BBLOCK ,		! addr of MVL entry for current vol
	UCB_LIST = 10	: REF VECTOR;		! addr of list of UCB's for vol set
```

```
2       EXTERNAL REGISTER
2            COMMON_REG;
2
2       LOCAL
2            MVL      : REF BBLOCK;                    ! address of MVL control block
2
2       ! get the MVL and see if we need to increase its size. This means that
2       ! if we have more volumes in the set then originally specified then we
2       ! must create more MVL enties for those volumes. Each volume in a volume
2       ! set has its own MVL for the duration of the mount of that volume set.
2
2       MVL = .CURRENT_VCB[VCB$L_MVL];
2       IF .MVL[MVL$B_NVOLS] LSS .VOL
2       THEN
2            MVL = KERNEL_CALL(MAKE_VOL_ENTRY, .VOL, .MVL);
2
2       ! point at the current MVL label
2
2       MVL_ENTRY = .MVL + MVL$K_FIXLEN + ((.VOL - 1)*MVL$K_LENGTH);
2
2       ! if volume mounted then make the volume and the unit it is mounted on
2       ! current. Else if the MTAACP is running in Automatic mode then all
2       ! we need to do is get the next free drive. We must assume that if
2       ! the drive has a vaild reel on it then it is the next reel the
2       ! operator wishes us to use. If we are not running in Automatic mode this
2       ! is not true and we must choose a unit, clear its previous use, and
2       ! make the volume and the new unit current.
2
2       UCB_LIST = BBLOCK[.CURRENT_VCB[VCB$L_RVT], RVT$L_UCBLST];
2       IF .MVL_ENTRY[MVL$V_MOUNTED] AND NOT .FLAGS[MOU$V_MOUNTERR]
3       THEN KERNEL_CALL(MAKE_CUR_VOL, .MVL_ENTRY[MVL$B_RVN], .VOL)
2       ELSE
3       BEGIN
3
3            ! If we are running in Automatic mode then we want to unload the
3            ! last volume so that the operator can put the next reel on the drive.
3            ! However we also want to special case the fact that the user
3            ! may have only one drive and thus force the operator to intervene.
3
3            IF NOT .CURRENT_VCB[VCB$V_NOAUTO]
4               AND (.BBLOCK [.CURRENT_VCB[VCB$L_RVT], RVT$B_NVOLS] GTR 1)
3            THEN
4            BEGIN
4               KERNEL_CALL(CLEAR_UNIT);
4               KERNEL_CALL(MAKE_CUR_VOL, CHOOSE_UNIT(), .VOL);
4            END
3            ELSE KERNEL_CALL(CLPREV_MAKECUR, CHOOSE_UNIT(), .VOL);
2       END;
2
2       ! now if the volume is mounted and no rewind is required just return
2
2       IF   .MVL_ENTRY[MVL$V_MOUNTED]
2            AND NOT .FLAGS[MOU$V_MOUNTERR]
2            AND NOT .FLAGS[MOU$V_REWIND]
2                THEN RETURN .MVL_ENTRY;
2
```

```
2       ! Assume that we won't send a message to the operators console. We would
2       ! want to sent one if we switched to a new reel on a volume set without
2       ! the operator getting involved in anyway. That is automatic volume
2       ! recognition is turned on and no errors occured while mounting the
2       ! next volume. Also assume that no label will be specified.
2
2       INFORM_OPER [0] = FALSE;
2       LABEL_SPEC [0] = FALSE;
2
2       ! Call to get the device name and length of the name. These fields are
2       ! stored in fields accessible to other routines in this module so that
2       ! only one call need be done.
2
2       GET_DEV_NAME (CVT_DEVNAM_LENGTH, CVT_DEVNAM);
2
2       ! When the Mtaacp is running in Automtic mode, before asking the
2       ! operator for the tape and label try to generate the label with
2       ! in the ACP. And try to mount the volume on the given unit with
2       ! that label.
2
2       IF NOT .CURRENT_VCB[VCB$V_NOAUTO]
2       THEN
3       BEGIN
4           IF KERNEL_CALL(CREATE_LABEL, .VOL, .MVL)
3             THEN
4             BEGIN
4
4                   ! Default the following fields
4                   ! LABEL_SPEC : We have a label for the next volume to mount.
4                   ! INFORM_OPER : If this mount works inform the operator that
4                   !               we have mounted a volume "behind his back".
4
4                   LABEL_SPEC [0] = TRUE;
4                   INFORM_OPER [0] = TRUE;
4
4                   ! If there is only one drive in this volume set then must give
4                   ! the operator some time to put the tape on the drive.
4
4                   IF .BBLOCK [.CURRENT_VCB[VCB$L_RVT], RVT$B_NVOLS] GTR 1
4                   THEN
5                   BEGIN
5                      KERNEL_CALL(ASSUME_MOUNTED);
5                      KERNEL_CALL(SEND_ERRLOG,1,.CURRENT_UCB);
4                   END;
3             END;
2       END;
2
2       ! loop until we have a good mount
2
2       WHILE 1 DO
3           BEGIN
3
3           LOCAL  STATUS;
3
3           ! assume all is going to work
3
3           STATUS = TRUE;
```

```
    ! Does the operator need to get involved ( mount a reel on a drive )

    IF NOT .MVL_ENTRY[MVL$V_MOUNTED] OR .FLAGS[MOU$V_MOUNTERR]
    THEN
        BEGIN

        LOCAL
            LABEL_ADDR : REF VECTOR [,BYTE],
            LABEL_SZ,
            MESSAGE_NUMBER;

        ! find the size to the label, do not print trailing spaces
        !
        LABEL_ADDR = MVL_ENTRY [ MVL$T_VOLLBL ];
        IF .MVL_ENTRY [MVL$V_UNUSED]
        THEN LABEL_SZ = 0
        ELSE
            BEGIN
            LABEL_SZ = 6;
            DECR I FROM MVL$S_VOLLBL - 1 TO 0 DO
                IF .LABEL_ADDR [.I] NEQ ' '
                THEN EXITLOOP
                ELSE LABEL_SZ = .LABEL_SZ - 1;
            END;

        ! tell the operator to mount the reel
        !
        IF .FLAGS[MOU$V_LBLCHECK]
        THEN MESSAGE_NUMBER = MOUN$_REMOUVOL
        ELSE IF .FLAGS[MOU$V_CHKIFSPC]
             THEN MESSAGE_NUMBER = MOUN$_MOUVOL
             ELSE MESSAGE_NUMBER = MOUN$_MOUNEWVOL;
        IF NOT PRINT_OPR_MSG(.MESSAGE_NUMBER, .MAIL_CHANNEL,
                .CURRENT_VCB[VCB$B_CUR_RVN], .LABEL_SZ, .LABEL_ADDR,
                .CVT_DEVNAM_LENGTH, CVT_DEVNAM)
        THEN
            BEGIN
            KERNEL_CALL(TERMINATE_VOL, .CURRENT_WCB);
            ERR_EXIT(SS$_NOTAPEOP);
            END;

        ! block the process and wait for the operator to reply
        !
        ENABLE_MAIL_AST();
        BLOCK(VCB$M_WAIMOUVOL);

        ! The operator has replied so check if a label has been specified
        ! If it has then check it to be sure that it is a valid ANSI label.

        IF .MAILSZ GTR 0
        THEN
            BEGIN

            ! stuff the label in MVL after doing ANSI checks

            STATUS = KERNEL_CALL(OPERATOR_LBL);
```

```
5           IF NOT .STATUS
5           THEN PRINT_OPR_MSG ( .STATUS, 0,
5                        .CVT_DEVNAM_LENGTH, CVT_DEVNAM)
5           ELSE LABEL_SPEC [0] = TRUE;
4           END;
4
4       ! No need to send another message to the operators console
4
4       INFORM_OPER [0] = FALSE;
4
4       ! assume device is mounted
4
4       KERNEL_CALL(ASSUME_MOUNTED);
4       KERNEL_CALL(SEND_ERRLOG, 1, .CURRENT_UCB);
3       END;
3
3   ! the reel was just mount or was already mounted now check it for
3   ! being on online and valid
3   !
3   IF .STATUS
3   THEN
3
3       ! Rewind the reel
3       !
3       INCRU J FROM 0 TO 29 DO
4           BEGIN
4           STATUS = REWIND_AND_WAIT();
4
4           ! if on_line, then exit loop
4           !
4           IF .STATUS THEN EXITLOOP;
4
4           ! wait one second if offline
4           !
4           IF SYS$SETIMR(TIMEFN, SECONDS, 0, 0)
4           THEN SYS$WAITFR(TIMEFN);
3           END;
3
3   ! check for the write ring if needed
3   !
4   IF .STATUS AND ( NOT .BBLOCK[CURRENT_UCB[UCB$L_DEVCHAR],DEV$V_SWL] )
4   THEN IF NOT (STATUS = CHECK_RING ())
3        THEN PRINT_OPR_MSG ( MOUN$_WRITLCK, 0,
3                        .CVT_DEVNAM_LENGTH, CVT_DEVNAM);
3
3   ! check the users privileges to write and read to the volume
3   !
3   IF .STATUS
3   THEN
4       BEGIN
4
4       ! assume device is mounted
4       !
4       KERNEL_CALL(ASSUME_MOUNTED);
4
4       ! exit if "/BLANK" on the reply command on a write next volume
4       ! operation
```

```
4               !
6               IF ((.MAIL[OPC$W_MS_STATUS] EQL (OPC$_BLANKTAPE AND %X'FFFF'))
5                  OR .CURRENT_VCB[VCB$V_BLANK] )
5                  AND NOT ( .FLAGS [MOU$V_CHKIFSPC] OR .FLAGS [MOU$V_LBLCHECK] )
4               THEN
5                   BEGIN
5
5                   ! if the use writes the tape he has override privs
5                   !
5                   KERNEL_CALL( SET_MVL_OVERIDE, TRUE);
5
5                   ! mount has succeeded exit "try till good mount" loop
5                   !
5                   EXITLOOP;
4                   END;
4
4               ! now check for ANSI accessiblity and VMS protection and
4               ! exit the "try till good mount" loop in everything is OK
4               !
4               IF CHECK_ACCESS ( .FLAGS ) THEN EXITLOOP;
4
3               END;
3
3           ! mount did not work for some reason, force operator intervention
3           !
3           FLAGS = .FLAGS OR $FIELDMASK(MOU$V_MOUNTERR);
3
3           ! reset the state of things
3           !
3           KERNEL_CALL(CLPREV_MAKECUR, .MVL_ENTRY[MVL$B_RVN], .VOL);
2           END;                                        ! end of while not good mount
2
2   ! Check to see if the operator should hear about the switch then return.
2
2   IF .INFORM_OPER [0]
2       THEN
3       BEGIN
3           LOCAL
3           DESCR : VECTOR [2];                     ! Descr of the device name for
3                                                   ! the FAO field in the msg
3
3           DESCR [0] = .CVT_DEVNAM_LENGTH;         ! Length of dev name
3           DESCR [1] = CVT_DEVNAM;                 ! Address of the dev name
3
3           ! Assume that the size of the label is 6. This is a safe assumption
3           ! because we generated the label.
3
3           PRINT_OPR_MSG (MOUN$_MOUNTED, 0, 6, MVL_ENTRY[MVL$T_VOLLBL],DESCR);
2         END;
2   RETURN .MVL_ENTRY;
2
1   END;                                            ! end of routine MOUNT_VOL

                                                     .TITLE  MOUVOL
                                                     .IDENT  \V04-000\
                                                                  .PSECT  $CODE$,NOWRT,2

   00  00  41  31  31  45  4C  49  46  43  45  44  00000 P.AAA:   .ASCII  \DECFILE11A\<0><0>                          ;
                             FFFFFFFF  FF676980  0000C P.AAB:   .LONG   -10000000, -1                               ;

                                                                  .PSECT  $LOCKEDD1$,NOEXE,2

                                                 00000 TAPE_OWNER:
                                                                  .BLKB   4
                                                 00004 TAPE_PROT:
                                                                  .BLKB   2
                                                 00006            .BLKB   2
                                                 00008 LABEL_SPEC:
                                                                  .BLKB   1
                                                 00009            .BLKB   3
                                                 0000C INFORM_OPER:
                                                                  .BLKB   1
                                                 0000D            .BLKB   3
                                                 00010 CVT_DEVNAM:
                                                                  .BLKB   16
                                                 00020 CVT_DEVNAM_LENGTH:
                                                                  .BLKB   1

                                                       STARID=            P.AAA
                                                       SECONDS=           P.AAB
                                                                  .EXTRN  CURRENT_UCB, CURRENT_WCB
                                                                  .EXTRN  IO_CHANNEL, IO_STATUS
                                                                  .EXTRN  MAIL_CHANNEL, WORK_AREA
                                                                  .EXTRN  BLOCK, CHECK_PROT
                                                                  .EXTRN  ENABLE_MAIL_AST
                                                                  .EXTRN  GET_CCB, GET_RECORD
                                                                  .EXTRN  IOC$CVT_DEVNAM, PRINT_OPR_MSG
                                                                  .EXTRN  PRINT_NOT_LABEL
                                                                  .EXTRN  PROCESS_VOL2_LABEL
                                                                  .EXTRN  READ_BLOCK, REWIND_AND_WAIT
                                                                  .EXTRN  SEND_ERRLOG, SYS$FAO
                                                                  .EXTRN  SYS$QIOW, SYS$SETIMR
                                                                  .EXTRN  SYS$WAITFR, TAPE_OWN_PROT
                                                                  .EXTRN  TERMINATE_VOL, SYS$CMKRNL

                                                                  .PSECT  $CODE$,NOWRT,2

                                            07FC 00000            .ENTRY  MOUNT_VOL, Save R2,R3,R4,R5,R6,R7,R8,R9,R10 ; 0723
                         58       0000V  CF   9E 00002            MOVAB   ASSUME_MOUNTED, R8                          ;
                         57       0000G  CF   9E 00007            MOVAB   PRINT_OPR_MSG, R7                           ;
                         56       0000'  CF   9E 0000C            MOVAB   CVT_DEVNAM, R6                              ;
                         55   00000000G  9F   9E 00011            MOVAB   @#SYS$CMKRNL, R5                            ;
                         5E              08   C2 00018            SUBL2   #8, SP                                      ;
                         54          34  AB   D0 0001B            MOVL    52(CURRENT_VCB), MVL                        ; 0792
                         52          04  AC   D0 0001F            MOVL    VOL, R2                                     ; 0793
   52        08   A4     08              00   ED 00023            CMPZV   #0, #8, 11(MVL), R2                         ;
                                         10   18 00029            BGEQ    1$                                          ;
                                         14   BB 0002B            PUSHR   #^M<R2,R4>                                  ; 0795
                                         02   DD 0002D            PUSHL   #2                                          ;
                                         5E   DD 0002F            PUSHL   SP                                          ;
                                  0000V  CF   9F 00031            PUSHAB  MAKE_VOL_ENTRY                              ;
                         65              05   FB 00035            CALLS   #5, SYS$CMKRNL                              ;
```

```
MOUVOL                                                    I 10
V04-000                                         16-Sep-1984 02:25:33    VAX-11 Bliss-32 V4.0-742            Page  12
                                                14-Sep-1984 12:46:44    [MTAACP.SRC]MOUVOL.B32;1                 (2)

                    54                50  D0 00038           MOVL    R0, MVL                              :
                    59          1C A442  7E 0003B  1$:       MOVAQ   28(MVL)[R2], MVL_ENTRY               : 0799
                    53          20   AB  D0 00040           MOVL    32(CURRENT_VCB), R3                  : 0809
                    5A          44   A3  9E 00044           MOVAB   68(R3), UCB_LIST                     :
                    0D          07   A9  E9 00048           BLBC    7(MVL_ENTRY), 2$                     : 0810
        08     08   AC               03  E0 0004C           BBS     #3, FLAGS, 2$                        :
                                     52  DD 00051           PUSHL   R2                                   : 0811
                    7E          06   A9  9A 00053           MOVZBL  6(MVL_ENTRY), -(SP)                  :
                                     1D  11 00057           BRB     3$                                   :
        22     2D   AB               04  E0 00059  2$:       BBS     #4, 45(CURRENT_VCB), 4$              : 0820
                    01          0B   A3  91 0005E           CMPB    11(R3), #1                           : 0821
                                     1C  1B 00062           BLEQU   4$                                   :
                                     7E  D4 00064           CLRL    -(SP)                                : 0824
                                     5E  DD 00066           PUSHL   SP                                   :
                             0000V   CF  9F 00068           PUSHAB  CLEAR_UNIT                           :
                    65               03  FB 0006C           CALLS   #3, SYS$CMKRNL                       :
                                     52  DD 0006F           PUSHL   R2                                   : 0825
                                  0000V  30 00071           BSBW    CHOOSE_UNIT                          :
                                     50  DD 00074           PUSHL   R0                                   :
                                     02  DD 00076  3$:       PUSHL   #2                                   :
                                     5E  DD 00078           PUSHL   SP                                   :
                             0000V   CF  9F 0007A           PUSHAB  MAKE_CUR_VOL                         :
                                     0F  11 0007E           BRB     5$                                   :
                                     52  DD 00080  4$:       PUSHL   R2                                   : 0827
                                  0000V  30 00082           BSBW    CHOOSE_UNIT                          :
                                     50  DD 00085           PUSHL   R0                                   :
                                     02  DD 00087           PUSHL   #2                                   :
                                     5E  DD 00089           PUSHL   SP                                   :
                             0000V   CF  9F 0008B           PUSHAB  CLPREV_MAKECUR                       :
                    65               05  FB 0008F  5$:       CALLS   #5, SYS$CMKRNL                       :
                    0C          07   A9  E9 00092           BLBC    7(MVL_ENTRY), 6$                     : 0832
        07     08   AC               03  E0 00096           BBS     #3, FLAGS, 6$                        : 0833
                    03          08   AC  E8 0009B           BLBS    FLAGS, 6$                            : 0834
                                  01F7   31 0009F           BRW     29$                                  :
               FC   A6               01  8A 000A2  6$:       BICB2   #1, INFORM_OPER                      : 0843
               F8   A6               01  8A 000A6           BICB2   #1, LABEL_SPEC                       : 0844
                                     56  DD 000AA           PUSHL   R6                                   : 0850
                                10   A6  9F 000AC           PUSHAB  CVT_DEVNAM_LENGTH                    :
            0000V   CF               02  FB 000AF           CALLS   #2, GET_DEV_NAME                     :
        3C     2D   AB               04  E0 000B4           BBS     #4, 45(CURRENT_VCB), 8$              : 0857
                                     14  BB 000B9           PUSHR   #^M<R2,R4>                           : 0860
                                     02  DD 000BB           PUSHL   #2                                   :
                                     5E  DD 000BD           PUSHL   SP                                   :
                             0000V   CF  9F 000BF           PUSHAB  CREATE_LABEL                         :
                    65               05  FB 000C3           CALLS   #5, SYS$CMKRNL                       :
                    2C               50  E9 000C6           BLBC    R0, 8$                               :
               F8   A6               01  88 000C9           BISB2   #1, LABEL_SPEC                       : 0869
               FC   A6               01  88 000CD           BISB2   #1, INFORM_OPER                      : 0870
                    50          20   AB  D0 000D1           MOVL    32(CURRENT_VCB), R0                  : 0875
                    01          0B   A0  91 000D5           CMPB    11(R0), #1                           :
                                     1A  1B 000D9           BLEQU   8$                                   :
                                     7E  D4 000DB           CLRL    -(SP)                                : 0878
                              4100   8F  BB 000DD           PUSHR   #^M<R8,SP>                           :
                    65               03  FB 000E1           CALLS   #3, SYS$CMKRNL                       :
                             0000G   CF  DD 000E4           PUSHL   CURRENT_UCB                          : 0879
                                     01  DD 000E8           PUSHL   #1                                   :
                                     02  DD 000EA           PUSHL   #2                                   :
```

```
                            5E  DD 000EC          PUSHL   SP                                       ;
                 0000G      CF  9F 000EE          PUSHAB  SEND_ERRLOG                              ;
              65            05  FB 000F2  7$:     CALLS   #5, SYS$CMKRNL                           ;
              53            01  D0 000F5  8$:     MOVL    #1, STATUS                               ; 0893
              08       07   A9  E9 000F8          BLBC    7(MVL_ENTRY), 9$                         ; 0897
     03       08   AC       03  E0 000FC          BBS     #3, FLAGS, 9$                            ;
                          00C5  31 00101          BRW     19$                                      ;
              50            59  D0 00104  9$:     MOVL    MVL_ENTRY, LABEL_ADDR                    ; 0908
     04       07   A9       01  E1 00107          BBC     #1, 7(MVL_ENTRY), 10$                    ; 0909
                            52  D4 0010C          CLRL    LABEL_SZ                                 ; 0910
                            11  11 0010E          BRB     12$                                      ;
              52            06  D0 00110  10$:    MOVL    #6, LABEL_SZ                             ; 0913
              51            05  D0 00113          MOVL    #5, I                                    ; 0915
              20          6140  91 00116  11$:    CMPB    (I)[LABEL_ADDR], #32                     ;
                            05  12 0011A          BNEQ    12$                                      ;
                            52  D7 0011C          DECL    LABEL_SZ                                 ; 0917
              F5            51  F4 0011E          SOBGEQ  I, 11$                                   ; 0915
     09       08   AC       01  E1 00121  12$:    BBC     #1, FLAGS, 13$                           ; 0922
              51 0072820C   8F  D0 00126          MOVL    #7504396, MESSAGE_NUMBER                 ; 0923
                            15  11 0012D          BRB     15$                                      ;
     09       08   AC       02  E1 0012F  13$:    BBC     #2, FLAGS, 14$                           ; 0924
              51 0072809C   8F  D0 00134          MOVL    #7504028, MESSAGE_NUMBER                 ; 0925
                            07  11 0013B          BRB     15$                                      ;
              51 00728204   8F  D0 0013D  14$:    MOVL    #7504388, MESSAGE_NUMBER                 ; 0926
                            56  DD 00144  15$:    PUSHL   R6                                       ; 0927
              7E       10   A6  9A 00146          MOVZBL  CVT_DEVNAM_LENGTH, -(SP)                 ; 0929
                            50  DD 0014A          PUSHL   LABEL_ADDR                               ; 0928
                            52  DD 0014C          PUSHL   LABEL_SZ                                 ;
              7E       2F   AB  9A 0014E          MOVZBL  47(CURRENT_VCB), -(SP)                   ;
                 0000G      CF  DD 00152          PUSHL   MAIL_CHANNEL                             ; 0927
                            51  DD 00156          PUSHL   MESSAGE_NUMBER                           ;
                            67  16 00158          JSB     PRINT_OPR_MSG                            ;
              5E            1C  C0 0015A          ADDL2   #28, SP                                  ;
              13            50  E8 0015D          BLBS    R0, 16$                                  ;
                 0000G      CF  DD 00160          PUSHL   CURRENT_WCB                              ; 0932
                            01  DD 00164          PUSHL   #1                                       ;
                            5E  DD 00166          PUSHL   SP                                       ;
                 0000G      CF  9F 00168          PUSHAB  TERMINATE_VOL                            ;
              65            04  FB 0016C          CALLS   #4, SYS$CMKRNL                           ;
                  0264      8F  BF 0016F          CHMU    #612                                     ; 0933
        0000G CF            00  FB 00173  16$:    CALLS   #0, ENABLE_MAIL_AST                      ; 0938
                            04  DD 00178          PUSHL   #4                                       ; 0939
        0000G CF            01  FB 0017A          CALLS   #1, BLOCK                                ;
                 0000G      CF  D5 0017F          TSTL    MAILSZ                                   ; 0944
                            26  15 00183          BLEQ    18$                                      ;
                            7E  D4 00185          CLRL    -(SP)                                    ; 0950
                            5E  DD 00187          PUSHL   SP                                       ;
                 0000V      CF  9F 00189          PUSHAB  OPERATOR_LBL                             ;
              65            03  FB 0018D          CALLS   #3, SYS$CMKRNL                           ;
              53            50  D0 00190          MOVL    R0, STATUS                               ;
              11            53  E8 00193          BLBS    STATUS, 17$                              ; 0951
                            56  DD 00196          PUSHL   R6                                       ; 0952
              7E       10   A6  9A 00198          MOVZBL  CVT_DEVNAM_LENGTH, -(SP)                 ; 0953
                            7E  D4 0019C          CLRL    -(SP)                                    ; 0952
                            53  DD 0019E          PUSHL   STATUS                                   ;
                            67  16 001A0          JSB     PRINT_OPR_MSG                            ;
              5E            10  C0 001A2          ADDL2   #16, SP                                  ;
```

```
                                         04  11 001A5        BRB     18$                               :
               F8    A6                  01  88 001A7 17$:   BISB2   #1, LABEL_SPEC                    : 0954
               FC    A6                  01  8A 001AB 18$:   BICB2   #1, INFORM_OPER                   : 0959
                                         7E  D4 001AF        CLRL    -(SP)                             : 0963
                           4100          8F  BB 001B1        PUSHR   #^M<R8,SP>                        :
                     65                  03  FB 001B5        CALLS   #3, SYS$CMKRNL                    :
                          0000G          CF  DD 001B8        PUSHL   CURRENT_UCB                       : 0964
                                         01  DD 001BC        PUSHL   #1                                :
                                         02  DD 001BE        PUSHL   #2                                :
                                         5E  DD 001C0        PUSHL   SP                                :
                          0000G          CF  9F 001C2        PUSHAB  SEND_ERRLOG                       :
                     65                  05  FB 001C6        CALLS   #5, SYS$CMKRNL                    :
                     5A                  53  E9 001C9 19$:   BLBC    STATUS, 23$                       : 0970
                                         52  D4 001CC        CLRL    J                                 : 0975
            0000G    CF                  00  FB 001CE 20$:   CALLS   #0, REWIND_AND_WAIT               : 0977
                     53                  50  D0 001D3        MOVL    R0, STATUS                        :
                     25                  53  E8 001D6        BLBS    STATUS, 22$                       : 0981
                                         7E  7C 001D9        CLRQ    -(SP)                             : 0985
                           FE19          CF  9F 001DB        PUSHAB  SECONDS                           :
                                         03  DD 001DF        PUSHL   #3                                :
        00000000G    9F                  04  FB 001E1        CALLS   #4, @#SYS$SETIMR                  :
                     09                  50  E9 001E8        BLBC    R0, 21$                           :
                                         03  DD 001EB        PUSHL   #3                                : 0986
        00000000G    9F                  01  FB 001ED        CALLS   #1, @#SYS$WAITFR                  :
                                         52  D6 001F4 21$:   INCL    J                                 : 0975
                     1D                  52  D1 001F6        CMPL    J, #29                            :
                                         D3  1B 001F9        BLEQU   20$                               :
                     66                  53  E9 001FB        BLBC    STATUS, 27$                       : 0991
                     50   0000G          CF  D0 001FE 22$:   MOVL    CURRENT_UCB, R0                   :
   1E         3B     A0                  01  E0 00203        BBS     #1, 59(R0), 23$                   :
            0000V    CF                  00  FB 00208        CALLS   #0, CHECK_RING                    : 0992
                     53                  50  D0 0020D        MOVL    R0, STATUS                        :
                     16                  53  E8 00210        BLBS    STATUS, 24$                       :
                                         56  DD 00213        PUSHL   R6                                : 0993
                     7E     10           A6  9A 00215        MOVZBL  CVT_DEVNAM_LENGTH, -(SP)          : 0994
                                         7E  D4 00219        CLRL    -(SP)                             : 0993
                       00728134          8F  DD 0021B        PUSHL   #7504180                          :
                                         67  16 00221        JSB     PRINT_OPR_MSG                     :
                     5E                  10  C0 00223        ADDL2   #16, SP                           :
                     3B                  53  E9 00226 23$:   BLBC    STATUS, 27$                       : 0998
                                         7E  D4 00229 24$:   CLRL    -(SP)                             : 1004
                           4100          8F  BB 0022B        PUSHR   #^M<R8,SP>                        :
                     65                  03  FB 0022F        CALLS   #3, SYS$CMKRNL                    :
             81E3    8F   0000G          CF  B1 00232        CMPW    MAIL+2, #33251                    : 1009
                                         05  13 00239        BEQL    25$                               :
   19         2D     AB                  02  E1 0023B        BBC     #2, 45(CURRENT_VCB), 26$          : 1010
   14         08     AC                  02  E0 00240 25$:   BBS     #2, FLAGS, 26$                    : 1011
   0F         08     AC                  01  E0 00245        BBS     #1, FLAGS, 26$                    :
                                         01  DD 0024A        PUSHL   #1                                : 1017
                                         01  DD 0024C        PUSHL   #1                                :
                                         5E  DD 0024E        PUSHL   SP                                :
                          0000V          CF  9F 00250        PUSHAB  SET_MVL_OVERIDE                   :
                     65                  04  FB 00254        CALLS   #4, SYS$CMKRNL                    :
                                         21  11 00257        BRB     28$                               : 1013
                             08          AC  DD 00259 26$:   PUSHL   FLAGS                             : 1027
            0000V    CF                  01  FB 0025C        CALLS   #1, CHECK_ACCESS                  :
                     16                  50  E8 00261        BLBS    R0, 28$                           :
```

```
MOUVOL                                            L 10
V04-000                                 16-Sep-1984 02:25:33    VAX-11 Bliss-32 V4.0-742           Page  15
                                        14-Sep-1984 12:46:44    [MTAACP.SRC]MOUVOL.B32;1                 (2)

                    08   AC            08  88 00264 27$:   BISB2   #8, FLAGS                            ; 1033
                               04      AC  DD 00268        PUSHL   VOL                                  ; 1037
                         7E    06      A9  9A 0026B        MOVZBL  6(MVL_ENTRY), -(SP)
                                       02  DD 0026F        PUSHL   #2
                                       5E  DD 00271        PUSHL   SP
                             0000V     CF  9F 00273        PUSHAB  CLPREV_MAKECUR
                                     FE78  31 00277        BRW     7$
                         1B    FC      A6  E9 0027A 28$:   BLBC    INFORM_OPER, 29$                     ; 1042
                         6E    10      A6  9A 0027E        MOVZBL  CVT_DEVNAM_LENGTH, DESCR             ; 1049
                    04   AE            66  9E 00282        MOVAB   CVT_DEVNAM, DESCR+4                  ; 1050
                             4200      8F  BB 00286        PUSHR   #^M<R9,SP>                           ; 1055
                                       06  DD 0028A        PUSHL   #6
                                       7E  D4 0028C        CLRL    -(SP)
                         0072A003      8F  DD 0028E        PUSHL   #7512067
                                       67  16 00294        JSB     PRINT_OPR_MSG
                         5E            14  C0 00296        ADDL2   #20, SP
                         50            59  D0 00299 29$:   MOVL    MVL_ENTRY, R0                        ; 1057
                                           04 0029C        RET                                          ; 1059

; Routine Size:  669 bytes,    Routine Base:  $CODE$ + 0014
```

```
GLOBAL ROUTINE GET_DEV_NAME(LEN, C_DEVN) : COMMON_CALL NOVALUE =

!++
!
! FUNCTIONAL DESCRIPTION:
!       This routine converts a UCB address into te device's name by calling
!       the system routine IOC$CVT_DEVANM.
!
! CALLING SEQUENCE:
!       GET_DEV_NAME(ARG1,ARG2)
!
! INPUT PARAMETERS:
!       ARG1 - Address to store the length of the device name
!       ARG2 - Address to store the name of the device
!
! IMPLICIT INPUTS:
!       CURRENT_UCB - address of current unit control block
!
! OUTPUT PARAMETERS:
!       The length of the device name and the device name are returned
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       NONE
!
! SIDE EFFECTS:
!       NONE
!
!--

    BEGIN

    EXTERNAL REGISTER
        COMMON_REG;

    BIND
        CVT_DEVNAME =  .C_DEVN,             ! Storage for cvted device name
        LENGTH      =  .LEN,                ! Length of device name
        IN_NAME_LENGTH = MAX_DEVNAM_LENGTH; ! Set the maxium lentgh if dev nam

    LOCAL
        OUT_NAME_LENGTH : BYTE,             ! Actual length of device name gotten from the convert
        CVT_DEVNAM_STATUS,                  ! Status of the convert on devnam
        DEV_NAME : VECTOR [MAX_DEVNAM_LENGTH,BYTE]; ! Local storage for the
                                                    ! name from the convert

    ! Call to the system routine to get the device name of the drive to use
    ! given the UCB associated with that drive. This routine gets both
    ! the node name and the device name.
    ! NOTE: DEV_NAME must be in local storage, because the routine expects
    ! this field to be always accessible.

    CVT_DEVNAM_STATUS = IOC$CVT_DEVNAM ( IN_NAME_LENGTH, DEV_NAME, 0,
                                         .CURRENT_UCB; OUT_NAME_LENGTH);

```

```
;  605         1117  2      ! Move the resultant string into a field accessable to the entire module.
;  606         1118  2      ! Also fill the first byte of the string with the size of the string for
;  607         1119  2      ! the call to PRINT_OPR_MSG. The FAO string expects the device name to be
;  608         1120  2      ! in this format.
;  609         1121  2
;  610         1122  2      CH$MOVE (.OUT_NAME_LENGTH,DEV_NAME,CVT_DEVNAME);
;  611         1123  2      LENGTH = .OUT_NAME_LENGTH;
;  612         1124  1      END;


                                                    IN_NAME_LENGTH=     16


                                          007C 00000        .ENTRY  GET_DEV_NAME, Save R2,R3,R4,R5,R6               ; 1060
                          5E           10  C2 00002         SUBL2   #16, SP                                         ;
                          51           6E  9E 00005         MOVAB   DEV_NAME, R1                                    ; 1114
                          55     0000G CF  D0 00008         MOVL    CURRENT_UCB, R5                                 ;
                                       54  D4 0000D         CLRL    R4                                              ;
                          50           10  D0 0000F         MOVL    #16, R0                                         ;
                           00000000G   9F  16 00012         JSB     @#IOC$CVT_DEVNAM                                ;
                          56           51  D0 00018         MOVL    R1, OUT_NAME_LENGTH                             ;
                          50           56  9A 0001B         MOVZBL  OUT_NAME_LENGTH, R0                             ; 1122
              08   BC     6E           50  28 0001E         MOVC3   R0, DEV_NAME, @C_DEVN                           ;
                     04   BC           56  9A 00023         MOVZBL  OUT_NAME_LENGTH, @LEN                           ; 1123
                                           04 00027         RET                                                     ; 1124

; Routine Size:  40 bytes,    Routine Base:  $CODE$ + 02B1


;  613         1125  1
;  614         1126  1
```

```
1 ROUTINE CHOOSE_UNIT : L$CHOOSE_UNIT =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       this routine chooses the next unit to use
1 !
1 ! CALLING SEQUENCE:
1 !       CHOOSE_UNIT()
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       CURRENT_VCB - address of current volume control block
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       relative unit number
1 !
1 ! SIDE EFFECTS:
1 !       The current algorythm is increment current unit.
1 !       If it is allocated then use it else wrap around to first unit.
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         COMMON_REG;
2
2     LOCAL
2         NUNITS,                               ! number of units allocated
2         RVT     : REF BBLOCK,                 ! address of unit table
2         UNIT;                                 ! unit to use
2
2     RVT = .CURRENT_VCB[VCB$L_RVT];            ! get address of unit table
2
2     NUNITS = .RVT[RVT$B_NVOLS];               ! get # of units allocated to
2                                               !  this volume set
2     UNIT = .CURRENT_VCB[VCB$W_RVN] + 1;       ! inc current rel unit number
2
2     IF .UNIT GEQ .NUNITS
2     THEN
2         UNIT = 0;                             ! if not allocated, use 1st one
2
2     RETURN .UNIT;
2
1     END;                                      ! end of routine CHOOSE_UNIT
```

```
                    50       20    AB  D0 00000 CHOOSE_UNIT:
                                                        MOVL    32(CURRENT_VCB), RVT                      ; 1168
                    51       0B    A0  9A 00004         MOVZBL  11(RVT), NUNITS                           ; 1170
                    50       0E    AB  3C 00008         MOVZWL  14(CURRENT_VCB), UNIT                     ; 1172
                                   50  D6 0000C         INCL    UNIT                                      ;
                    51             50  D1 0000E         CMPL    UNIT, NUNITS                              ; 1174
                                   02  19 00011         BLSS    1$                                        ;
                                   50  D4 00013         CLRL    UNIT                                      ; 1176
                                       05 00015 1$:     RSB                                               ; 1180

; Routine Size:  22 bytes,    Routine Base:  $CODE$ + 02D9
```

```
1 ROUTINE CLEAR_UNIT : NOVALUE MVL_UCB =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine dismounts the last used volume in a volume set. It does
1 !       this by checking the Current UCB against the list of UCB's to find the
1 !       unit number. Then checking each entry in the MVL for that unit number.
1 !       If that unit is marked mounted then a QIO is issued to dismount and
1 !       unload that volume and the MVL is marked dismounted. This will allow
1 !       the operator to mount the next reel in the volume set before it is
1 !       actually needed by MTAACP processing.
1 !
1 ! CALLING SEQUENCE:
1 !       CLEAR_UNIT() called in Kernel mode
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       CURRENT_VCB - address of current volume control block
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       Last mounted volume is dismount and unloaded.
1 !
1 ! ROUTINE VALUE:
1 !       none
1 !
1 ! SIDE EFFECTS:
1 !       none
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         MVL_ENTRY = 9 : REF BBLOCK,             ! Address of current MVL not
2                                                 ! used by this routine
2         UCB_LIST = 10 : REF VECTOR,             ! Addr of UCB list for vol set
2         COMMON_REG;
2
2     LOCAL
2         NUNITS,                                 ! number of units allocated
2         RVT     : REF BBLOCK,                   ! address of unit table
2         UNIT,                                   ! unit currently in use
2         NVOLS,                                  ! number of volumes in set
2         MVL     : REF BBLOCK,                   ! addr of magtape volume list
2         MVL_ADDR : REF BBLOCKVECTOR [10,MVL$K_LENGTH];
2                                                 ! addr of MVL control block
2
2
2     RVT = .CURRENT_VCB[VCB$L_RVT];              ! get address of unit table
2     NUNITS = .RVT[RVT$B_NVOLS];                 ! get # of units allocated to
```

```
;  728        1238  2      UNIT = 0;                                          ! Set first unit
;  729        1239  2
;  730        1240  2    ! Check thru the list of UCB to find the unit number of the UCB currently
;  731        1241  2    ! in use.
;  732        1242  2
;  733        1243  2      INCR I FROM 0 TO .NUNITS - 1 DO
;  734        1244  2          IF .UCB_LIST [ .UNIT ] EQL .CURRENT_UCB
;  735        1245  2          THEN EXITLOOP
;  736        1246  2          ELSE UNIT = .UNIT + 1;
;  737        1247  2      MVL = .CURRENT_VCB[VCB$L_MVL];                     ! get the magtape vol list addr
;  738        1248  2      NVOLS = .MVL[MVL$B_NVOLS];                         ! get the numb of volume in set
;  739        1249  2      MVL_ADDR = .MVL + MVL$K_FIXLEN;                    ! get the first volume entry
;  740        1250  2
;  741        1251  2    ! Check thru the mounted volume labels to find the entry which points
;  742        1252  2    ! at the current unit and is mark as mounted. When we find that entry,
;  743        1253  2    ! a message is send to the ERRORLOG saying the volume has been dismounted;
;  744        1254  2    ! the entry is marked dismounted and a QIO is issued to the drive to unload
;  745        1255  2    ! the reel from the drive.
;  746        1256  2
;  747        1257  2      INCR I FROM 0 TO .NVOLS - 1 DO
;  748        1258  2          IF .MVL_ADDR [ .I, MVL$V_MOUNTED ]
;  749        1259  2             AND
;  750        1260  2             .UNIT EQL .MVL_ADDR [ .I, MVL$B_RVN ]
;  751        1261  2          THEN
;  752        1262  3          BEGIN
;  753        1263  3             KERNEL_CALL(SEND_ERRLOG,0,.UCB_LIST[.UNIT]);
;  754        1264  3             MVL_ADDR [ .I, MVL$V_MOUNTED ] = 0;
;  755        1265  3             SYS$QIOW ( 0, .IO_CHANNEL,
;  756        1266  3                        IO$_UNLOAD OR IO$M_NOWAIT OR IO$M_CLSEREXCP,
;  757        1267  3                        0, 0, 0, 0, 0, 0, 0, 0, 0);
;  758        1268  3             EXITLOOP;
;  759        1269  2          END;
;  760        1270  1      END;



                                        003C 00000 CLEAR_UNIT:
                                                           .WORD   Save R2,R3,R4,R5                            ; 1181
                 50       20    AB  D0 00002               MOVL    32(CURRENT_VCB), RVT                        ; 1236
                 51       08    A0  9A 00006               MOVZBL  11(RVT), NUNITS                             ; 1237
                                54  D4 0000A               CLRL    UNIT                                        ; 1238
                 50             01  CE 0000C               MNEGL   #1, I                                       ; 1244
                                0A  11 0000F               BRB     2$                                          ;
          0000G  CF           6A44  D1 00011 1$:           CMPL    (UCB_LIST)[UNIT], CURRENT_UCB               ;
                                06  13 00017               BEQL    3$                                          ;
                                54  D6 00019               INCL    UNIT                                        ; 1246
   F2            50             51  F2 0001B 2$:           AOBLSS  NUNITS, I, 1$                               ; 1244
                 50       34    AB  D0 0001F 3$:           MOVL    52(CURRENT_VCB), MVL                        ; 1247
                 55       0B    A0  9A 00023               MOVZBL  11(MVL), NVOLS                              ; 1248
                 52       24    A0  9E 00027               MOVAB   36(R0), MVL_ADDR                            ; 1249
                 53             01  CE 0002B               MNEGL   #1, I                                       ; 1258
                                4A  11 0002E               BRB     5$                                          ;
                          07  A243  7F 00030 4$:           PUSHAQ  7(MVL_ADDR)[I]                              ;
   42            9E             00  E1 00034               BBC     #0, @(SP)+, 5$                              ;
                          06  A243  7F 00038               PUSHAQ  6(MVL_ADDR)[I]                              ; 1260
```

```
        54          9E          08              00  ED 0003C          CMPZV   #0, #8, @(SP)+, UNIT              ;
                                                37  12 00041          BNEQ    5$                                ;
                                              6A44  DD 00043          PUSHL   (UCB_LIST)[UNIT]                  ; 1263
                                7E              02  7D 00046          MOVQ    #2, -(SP)                         ;
                                                5E  DD 00049          PUSHL   SP                                ;
                                        0000G   CF  9F 0004B          PUSHAB  SEND_ERRLOG                       ;
                    00000000G   9F              05  FB 0004F          CALLS   #5, @#SYS$CMKRNL                  ;
                                          07 A243   7F 00056          PUSHAQ  7(MVL_ADDR)[I]                    ; 1264
                                9E              01  8A 0005A          BICB2   #1, @(SP)+                        ;
                                                7E  7C 0005D          CLRQ    -(SP)                             ; 1265
                                                7E  7C 0005F          CLRQ    -(SP)                             ;
                                                7E  7C 00061          CLRQ    -(SP)                             ;
                                                7E  7C 00063          CLRQ    -(SP)                             ;
                                                7E  D4 00065          CLRL    -(SP)                             ;
                                7E      0281    8F  3C 00067          MOVZWL  #641, -(SP)                       ; 1266
                                        0000G   CF  DD 0006C          PUSHL   IO_CHANNEL                        ; 1265
                                                7E  D4 00070          CLRL    -(SP)                             ;
                    00000000G   9F              0C  FB 00072          CALLS   #12, @#SYS$QIOW                   ;
                                                04  00079             RET                                       ; 1262
                    B2          53              55  F2 0007A 5$:      AOBLSS  NVOLS, I, 4$                      ; 1258
                                                04  0007E             RET                                       ; 1270

; Routine Size:  127 bytes,    Routine Base:  $CODE$ + 02EF


;  761          1271  1
```

```
1 ROUTINE CLPREV_MAKECUR (UNIT, VOL) : MVL_UCB NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine clears the previous use of this unit in the
1 !       mag tape volume list and makes the unit and rel volume current
1 !
1 ! CALLING SEQUENCE:
1 !       CLPREV_MAKECUR(ARG1,ARG2), CALLED IN KERNEL MODE
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1 - relative unit number
1 !       ARG2 - relative volume number
1 !
1 ! IMPLICIT INPUTS:
1 !       CURRENT_VCB - address of current volume control block
1 !       MVL_ENTRY - address of current entry in mvl for volume
1 !       UCB_LIST - address of ucb's available for this unit
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       if volume currently mounted on unit, rewind issued
1 !
1 !--
1
2    BEGIN
2
2    EXTERNAL REGISTER
2       MVL_ENTRY = 9   : REF BBLOCK,    ! address of current rel vol in mvl
2       UCB_LIST = 10   : REF VECTOR,    ! address of UCB's
2       COMMON_REG;
2
2    LOCAL
2       CCB             : REF BBLOCK,               ! addr of ACP IO channel
2                                                   !   control block
2       MVL             : REF BBLOCK,               ! addr magtape volume list
2       MVL_ADDR        : REF BBLOCKVECTOR [10, MVL$K_LENGTH],
2                                                   ! address of MVL entries
2       NVOLS;
2
2
2    CCB = GET_CCB ( .IO_CHANNEL );                 ! calc ACP IO CCB
2    MVL = .CURRENT_VCB[VCB$L_MVL];                 ! address of mvl control block
2    NVOLS = .MVL[MVL$B_NVOLS];                     ! number of relative volume
2    MVL_ADDR = .MVL + MVL$K_FIXLEN;                ! addr of entries for rel vol
2
2    INCR I FROM 0 TO .NVOLS - 1 DO                 ! check each relative volume
3       BEGIN
```

```
;  820        1329 3          IF .MVL_ADDR[.I, MVL$V_MOUNTED]
;  821        1330 3              AND
;  822        1331 3                                              ! if mounted
;  823        1332 3              .UNIT EQL .MVL_ADDR[.I, MVL$B_RVN]  ! on unit about to be used
;  824        1333 3          THEN
;  825        1334 3              BEGIN
;  826        1335 4
;  827        1336 4              KERNEL_CALL(SEND_ERRLOG, 0, .UCB_LIST[.UNIT]);
;  828        1337 4                                              ! before mounted is cleared!!
;  829        1338 4              MVL_ADDR[.I, MVL$V_MOUNTED] = 0;    ! mark it dismounted,
;  830        1339 4
;  831        1340 4              ! assign channel to it's unit,
;  832        1341 4              !
;  833        1342 4              CCB[CCB$L_UCB] = .UCB_LIST[.UNIT];
;  834        1343 4              SYS$QIOW(0, .IO_CHANNEL,
;  835        1344 4                      IO$_UNLOAD OR IO$M_NOWAIT OR IO$M_CLSEREXCP,
;  836        1345 4                      0,0,0,0,0,0,0,0,0 ); ! unload the tape, please
;  837        1346 4              EXITLOOP;
;  838        1347 4
;  839        1348 4              END;
;  840        1349 3
;  841        1350 3          END;
;  842        1351 2
;  843        1352 2      MAKE_CUR_VOL(.UNIT, .VOL);
;  844        1353 2      END;                                        ! end of routine CLPREV_MAKECUR
;  845        1354 1



                                          007C 00000 CLPREV_MAKECUR:
                                                             .WORD   Save R2,R3,R4,R5,R6              ; 1272
                               0000G   CF   DD 00002         PUSHL   IO_CHANNEL                       ; 1322
                    0000G   CF            01   FB 00006         CALLS   #1, GET_CCB                      ;
                            56            50   D0 0000B         MOVL    R0, CCB                          ;
                            50     34   AB   D0 0000E         MOVL    52(CURRENT_VCB), MVL             ; 1323
                            55     0B   A0   9A 00012         MOVZBL  11(MVL), NVOLS                   ; 1324
                            52     24   A0   9E 00016         MOVAB   36(R0), MVL_ADDR                 ; 1325
                            54            01   CE 0001A         MNEGL   #1, I                            ; 1330
                                          54   11 0001D         BRB     2$                               ;
                                 07 A244   7F 0001F  1$:     PUSHAQ  7(MVL_ADDR)[I]                   ;
                 4C         9E            00   E1 00023         BBC     #0, @(SP)+, 2$                   ;
                                 06 A244   7F 00027         PUSHAQ  6(MVL_ADDR)[I]                   ; 1333
   04   AC       9E         08            00   ED 0002B         CMPZV   #0, #8, @(SP)+, UNIT             ;
                                          40   12 00031         BNEQ    2$                               ;
                            53     04   AC   D0 00033         MOVL    UNIT, R3                         ; 1337
                                    6A43   DD 00037         PUSHL   (UCB_LIST)[R3]                   ;
                            7E            02   7D 0003A         MOVQ    #2, -(SP)                        ;
                                          5E   DD 0003D         PUSHL   SP                               ;
                               0000G   CF   9F 0003F         PUSHAB  SEND_ERRLOG                      ;
                000000000G  9F            05   FB 00043         CALLS   #5, @#SYS$CMKRNL                 ;
                                 07 A244   7F 0004A         PUSHAQ  7(MVL_ADDR)[I]                   ; 1339
                            9E            01   8A 0004E         BICB2   #1, @(SP)+                       ;
                            66      6A43   D0 00051         MOVL    (UCB_LIST)[R3], (CCB)            ; 1343
                                          7E   7C 00055         CLRQ    -(SP)                            ; 1344
                                          7E   7C 00057         CLRQ    -(SP)                            ;
```

```
                                                            I 11
MOUVOL                                          16-Sep-1984 02:25:33    VAX-11 Bliss-32 V4.0-742            Page  25
V04-000                                         14-Sep-1984 12:46:44    [MTAACP.SRC]MOUVOL.B32;1                  (6)

                                   7E  7C 00059         CLRQ    -(SP)                                  ;
                                   7E  7C 0005B         CLRQ    -(SP)                                  ;
                                   7E  D4 0005D         CLRL    -(SP)                                  ;
                 7E       0281     8F  3C 0005F         MOVZWL  #641, -(SP)                            ; 1345
                          0000G    CF  DD 00064         PUSHL   IO_CHANNEL                             ; 1344
                                   7E  D4 00068         CLRL    -(SP)                                  ;
         00000000G  9F             0C  FB 0006A         CALLS   #12, @#SYS$QIOW                        ;
                                   04  11 00071         BRB     3$                                     ; 1335
       A8           54             55  F2 00073 2$:     AOBLSS  NVOLS, I, 1$                           ; 1327
                    7E        04   AC  7D 00077 3$:     MOVQ    UNIT, -(SP)                            ; 1353
             0000V  CF             02  FB 0007B         CALLS   #2, MAKE_CUR_VOL                       ;
                                       04 00080         RET                                            ; 1354

; Routine Size:  129 bytes,    Routine Base:  $CODE$ + 036E
```

```
ROUTINE MAKE_CUR_VOL (UNIT, VOL) : NOVALUE MVL_UCB =

!++
!
! FUNCTIONAL DESCRIPTION:
!       This routine makes the relative volume number on the rel unit the
!       current volume it notes that the current volume is mounted on this
!       relative unit
!
! CALLING SEQUENCE:
!       MAKE_CUR_VOL(ARG1,ARG2) in kernel mode
!
! INPUT PARAMETERS:
!       ARG1 - relative unit number on which the relative volume is mounted
!       ARG2 - relative volume number(starts at 1)
!
! IMPLICIT INPUTS:
!       NONE
!
! OUTPUT PARAMETERS:
!       NONE
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       NONE
!
! SIDE EFFECTS:
!       NONE
!
!--

    BEGIN

    EXTERNAL REGISTER
        MVL_ENTRY = 9   : REF BBLOCK,   ! address of current rel vol in mvl
        UCB_LIST = 10   : REF VECTOR,   ! address of ucb's allocated
        COMMON_REG;

    LOCAL
        CCB     : REF BBLOCK;           ! address of channel control block

    ! calculate address of channel control block
    !
    CCB = GET_CCB ( .IO_CHANNEL );

    ! assign channel to unit and set current ucb
    !
    CCB[CCB$L_UCB] = .UCB_LIST[.UNIT];
    CURRENT_UCB = .UCB_LIST[.UNIT];

    ! now set volume control block fields
    !
    CURRENT_VCB[VCB$W_RVN] = .UNIT;
    CURRENT_VCB[VCB$B_CUR_RVN] = .VOL;
```

```
;  904          1412  2     ! note which unit the volume is mounted on
;  905          1413  2     !
;  906          1414  2     MVL_ENTRY [ MVL$B_RVN ] = .UNIT;
;  907          1415  1     END;                                        ! end of routine MAKE_CUR_VOL



                                          0000 00000 MAKE_CUR_VOL:
                                                          .WORD   Save nothing                              ; 1355
                               0000G  CF   DD 00002       PUSHL   IO_CHANNEL                                ; 1400
                 0000G  CF            01   FB 00006       CALLS   #1, GET_CCB                               ;
                        51       04   AC   D0 0000B       MOVL    UNIT, R1                                  ; 1404
                        60          6A41   D0 0000F       MOVL    (UCB_LIST)[R1], (CCB)                     ;
                 0000G  CF          6A41   D0 00013       MOVL    (UCB_LIST)[R1], CURRENT_UCB               ; 1405
                    0E  AB            51   B0 00019       MOVW    R1, 14(CURRENT_VCB)                       ; 1409
                    2F  AB       08   AC   90 0001D       MOVB    VOL, 47(CURRENT_VCB)                      ; 1410
                    06  A9            51   90 00022       MOVB    R1, 6(MVL_ENTRY)                          ; 1414
                                           04 00026       RET                                               ; 1415

; Routine Size:  39 bytes,    Routine Base:  $CODE$ + 03EF
```

```
1 ROUTINE MAKE_VOL_ENTRY (VOL, MVL) : COMMON_CALL =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine puts a relative volume in the magnetic tape volume
1 !       list by making a new block and deallocating the old one
1 !
1 ! CALLING SEQUENCE:
1 !       MAKE_VOL_ENTRY(ARG1,ARG2), CALLED IN KERNEL MODE
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1 - volume number
1 !       ARG2 - address of magnetic tape volume list
1 !
1 ! IMPLICIT INPUTS:
1 !       CURRENT_VCB - address of current volume control block
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       MVL rebuilt
1 !
1 ! ROUTINE VALUE:
1 !       address of mvl control block
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    EXTERNAL REGISTER
2        COMMON_REG;
2
2    MAP
2        MVL     : REF BBLOCK;                   ! addr of mag tape volume list
2
2    LOCAL
2
2        MVL_ADDR: REF BBLOCKVECTOR [10, MVL$K_LENGTH],
2                                                ! address of MVL control block
2        NEWMVL  : REF BBLOCK,                   ! address of new MVL
2        NVOL;                                   ! number of volumes
2
2    EXTERNAL ROUTINE
2        ALLOCATE,                               ! allocate non_paged system mem
2        DEALLOCATE;                             ! deallocate non_paged sys mem
2
2    VOL  = .VOL + 4;                            ! grab some extra slots
2    NVOL = .MVL[MVL$B_NVOLS];                   ! get # of vols it will hold
2    NEWMVL = ALLOCATE((.VOL*MVL$K_LENGTH) + MVL$K_FIXLEN);
2                                                ! allocate non_paged sys space
2
2     ! initialize new MVL
```

```
;  966          1473 2      !
;  967          1474 2      NEWMVL[MVL$B_TYPE]  = DYN$C_MVL;
;  968          1475 2      NEWMVL[MVL$L_VCB]   = .CURRENT_VCB;
;  969          1476 2      NEWMVL[MVL$B_NVOLS] = .VOL;
;  970          1477 2
;  971          1478 2      ! copy all the old volume labels, File-Set-ID, and Vol_Acc
;  972          1479 2      !
;  973          1480 2      CH$MOVE(.MVL[MVL$W_SIZE] - 12, .MVL + 12, .NEWMVL + 12);
;  974          1481 2
;  975          1482 2      ! blank new relative volume lables
;  976          1483 2      !
;  977          1484 2      MVL_ADDR = .NEWMVL + MVL$K_FIXLEN;
;  978          1485 2      INCR I FROM .NVOL TO .VOL - 1 DO
;  979          1486 3          BEGIN
;  980          1487 3          CH$FILL(' ', MVL$S_VOLLBL, MVL_ADDR[.I, MVL$T_VOLLBL]);
;  981          1488 3          MVL_ADDR [ .I, MVL$V_UNUSED  ] = 1;
;  982          1489 3          MVL_ADDR [ .I, MVL$V_MOUNTED ] = 0;
;  983          1490 2          END;
;  984          1491 2
;  985          1492 2      ! set pointers to the new
;  986          1493 2      !
;  987          1494 2      CURRENT_VCB[VCB$L_MVL] = .NEWMVL;
;  988          1495 2
;  989          1496 2      ! get rid of the old
;  990          1497 2      !
;  991          1498 2      DEALLOCATE(.MVL);
;  992          1499 2
;  993          1500 2      RETURN .NEWMVL;
;  994          1501 2
;  995          1502 1      END;                                        ! end of routine MAKE_VOL_ENTRY


                                                                .EXTRN  ALLOCATE, DEALLOCATE

                                           07FC 00000 MAKE_VOL_ENTRY:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10              ; 1416
                         5E           04   C2 00002             SUBL2   #4, SP                                        ;
                   04    AC           04   C0 00005             ADDL2   #4, VOL                                       ; 1467
                         59     08    AC   D0 00009             MOVL    MVL, R9                                       ; 1468
                         56     0B    A9   9A 0000D             MOVZBL  11(R9), NVOL                                  ;
                         5A     04    AC   D0 00011             MOVL    VOL, R10                                      ; 1469
            7E           5A           03   78 00015             ASHL    #3, R10, -(SP)                                ;
                         6E           24   C0 00019             ADDL2   #36, (SP)                                     ;
                 0000G   CF           01   FB 0001C             CALLS   #1, ALLOCATE                                  ;
                         58           50   D0 00021             MOVL    R0, NEWMVL                                    ;
                   0A    A8           16   90 00024             MOVB    #22, 10(NEWMVL)                               ; 1474
                         68           5B   D0 00028             MOVL    CURRENT_VCB, (NEWMVL)                         ; 1475
                   0B    A8           5A   90 0002B             MOVB    R10, 11(NEWMVL)                               ; 1476
                         50     08    A9   3C 0002F             MOVZWL  8(R9), R0                                     ; 1480
                         50           0C   C2 00033             SUBL2   #12, R0                                       ;
       0C   A8     0C    A9           50   28 00036             MOVC3   R0, 12(R9), 12(NEWMVL)                        ;
                         57     24    A8   9E 0003C             MOVAB   36(R8), MVL_ADDR                              ; 1484
                                      56   D7 00040             DECL    I                                             ; 1487
                                      19   11 00042             BRB     2$                                            ;
                         6E         6746   7E 00044 1$:         MOVAQ   (MVL_ADDR)[I], (SP)                           ;
  06        20           6E           00   2C 00048             MOVC5   #0, (SP), #32, #6, @0(SP)                     ;
```

```
                                   00    BE      0004D                                                         :
                                   07 A746    7F 0004F           PUSHAQ   7(MVL_ADDR)[I]                       : 1488
                           9E         02    88 00053           BISB2    #2, @(SP)+                           :
                                   07 A746    7F 00056           PUSHAQ   7(MVL_ADDR)[I]                       : 1489
                           9E         01    8A 0005A           BICB2    #1, @(SP)+                           :
               E3          56         5A    F2 0005D  2$:      AOBLSS   R10, I, 1$                           : 1485
                      34   AB         58    D0 00061           MOVL     NEWMVL, 52(CURRENT_VCB)              : 1494
                                      59    DD 00065           PUSHL    R9                                   : 1498
                   0000G   CF         01    FB 00067           CALLS    #1, DEALLOCATE                       :
                           50         58    D0 0006C           MOVL     NEWMVL, R0                           : 1500
                                            04 0006F           RET                                           : 1502

; Routine Size:  112 bytes,    Routine Base:  $CODE$ + 0416
```

```
1 ROUTINE CREATE_LABEL (VOL, MVL) : COMMON_CALL =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine will create a label for a given volume if it does not
1 !       already have a label. It does this by getting the first four characters
1 !       of the label from the MVL of the previous reel in the volume set;
1 !       padding the blanks with the underscore character and putting in the
1 !       RVN of the current reel as the last two chars. This means
1 !       that only up to 99 different labels can be generate for a given volume
1 !       set.
1 !
1 !
1 ! CALLING SEQUENCE:
1 !       CREATE_LABEL(ARG1,ARG2), CALLED IN KERNEL MODE
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1 - volume number
1 !       ARG2 - address of magnetic tape volume list
1 !
1 ! IMPLICIT INPUTS:
1 !       CURRENT_VCB - address of current volume control block
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       MVL for the current reel is given a Volume Label
1 !
1 ! ROUTINE VALUE:
1 !       True : The label could be generated
1 !       False : The label could not be generated
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2   BEGIN
2
2   EXTERNAL REGISTER
2       COMMON_REG;
2
2   MAP
2       MVL     : REF BBLOCK;                   ! addr of mag tape volume list
2
2   LOCAL
2
2       CUR_MVL_ADDR: REF BBLOCK,
2                                               ! address of MVL control block
2       PRE_MVL_ADDR: REF BBLOCK,
2                                               ! address of MVL control block
2       LABEL_ADDR : REF VECTOR [,BYTE],        ! Place in label top reel numb
2       NUMBER_OF_TAPE,                         ! Number of current reel mod 99
2       DESCR : VECTOR [2];                     ! Descr for FAO string
2
```

```
        2 ! Get the number of he previous  volume
        2 ! and if that volume number is less than 0 then the current volume we
        2 ! are using must be the first volume in the reel
        2 !
        2       IF (.VOL-1) LEQ 0 THEN RETURN FALSE;
        2
        2 ! Get the individual MVL portion of the MVL block
        2 ! and move he first four chars of the previous reel's label into
        2 ! the current reels' label
        2
        2       PRE_MVL_ADDR = .MVL + MVL$K_FIXLEN + ((( .VOL-1 ) - 1) * MVL$K_LENGTH);
        2       CUR_MVL_ADDR = .MVL + MVL$K_FIXLEN + ((  .VOL-1 ) * MVL$K_LENGTH);
        2       CH$MOVE ((MVL$S_VOLLBL-2), PRE_MVL_ADDR [ MVL$T_VOLLBL ],
        2                  CUR_MVL_ADDR [ MVL$T_VOLLBL ]);
        2
        2 ! Check thru the new label for blanks and overwrite them with the underscore
        2 ! character.
        2
        2       LABEL_ADDR = CUR_MVL_ADDR [ MVL$T_VOLLBL ];
        2       DECR I FROM MVL$S_VOLLBL TO 2 DO
        2          IF .LABEL_ADDR [ .I ] EQL ' '
        2          THEN LABEL_ADDR [ .I ] = %C'_';
        2
        2 ! Now check to see if the RVN of this reel is greater then 99 decimal. If
        2 ! it is then set it to the RVN modulo 99.
        2
        2       NUMBER_OF_TAPE = .VOL;
        2       IF .NUMBER_OF_TAPE GTR 99
        2         THEN NUMBER_OF_TAPE = (.NUMBER_OF_TAPE MOD 99);
        2
        2 ! Set up the descriptors for the call to FAO and call FAO to convert the number
        2 ! to an ASCII string and insert it into the label field in the MVL of the
        2 ! current volume. Then set that the MVL is used and return to caller.
        2
        2       DESCR[0] = 2;                                          ! Size of output buffer
        2       DESCR[1] = CUR_MVL_ADDR [ MVL$T_VOLLBL ] + (MVL$S_VOLLBL-2);
        2                                                              ! Addr of output buffer
        2       SYS$FAO (DESCRIPTOR ('!2ZB'),
        2               0,
        2               DESCR,
        2               .NUMBER_OF_TAPE);
        2       CUR_MVL_ADDR[ MVL$V_UNUSED ] = 0;
        2       RETURN TRUE;
        1    END;


                        42  5A  32  21  00486 P.AAD:   .ASCII   \!2ZB\                             ;
                                        0048A          .BLKB    2
                             00000004  0048C P.AAC:   .LONG    4                                  ;
                             00000000' 00490          .ADDRESS P.AAD                             ;


                                   0004 00000 CREATE_LABEL:
                                                       .WORD    Save R2                            ; 1503
                   5E              08  C2 00002        SUBL2    #8, SP                             ;
```

```
MOUVOL                                                    D 12
V04-000                                      16-Sep-1984 02:25:33    VAX-11 Bliss-32 V4.0-742                 Page  33
                                             14-Sep-1984 12:46:44    [MTAACP.SRC]MOUVOL.B32;1                      (9)

                                  01      04   AC  D1 00005         CMPL    VOL, #1                                   ; 1564
                                               6B  15 00009         BLEQ    4$                                        ;
                                  50      04   AC  D0 0000B         MOVL    VOL, R0                                   ; 1570
                                  51   08 BC40 7E  0000F            MOVAQ   @MVL[R0], PRE_MVL_ADDR                    ;
                                  51           14  C0 00014         ADDL2   #20, PRE_MVL_ADDR                         ;
                                  52   08 BC40 7E  00017            MOVAQ   @MVL[R0], CUR_MVL_ADDR                    ; 1571
                                  52           1C  C0 0001C         ADDL2   #28, CUR_MVL_ADDR                         ;
                                  62           61  D0 0001F         MOVL    (PRE_MVL_ADDR), (CUR_MVL_ADDR)            ; 1573
                                  50           52  D0 00022         MOVL    CUR_MVL_ADDR, LABEL_ADDR                  ; 1578
                                  51           06  D0 00025         MOVL    #6, I                                     ; 1580
                                  20         6140  91 00028 1$:     CMPB    (I)[LABEL_ADDR], #32                      ;
                                               05  12 0002C         BNEQ    2$                                        ;
                                6140      5F   8F  90 0002E         MOVB    #95, (I)[LABEL_ADDR]                      ; 1581
         FFEE         51      FF  8F           02  9D 00033 2$:     ACBB    #2, #-1, I, 1$                            ; 1580
                                  50      04   AC  D0 0003A         MOVL    VOL, NUMBER_OF_TAPE                       ; 1586
                        00000063  8F           50  D1 0003E         CMPL    NUMBER_OF_TAPE, #99                       ; 1587
                                               0E  15 00045         BLEQ    3$                                        ;
          7E          00          50           01  7A 00047         EMUL    #1, NUMBER_OF_TAPE, #0, -(SP)             ; 1588
          50          50          8E 00000063  8F  7B 0004C         EDIV    #99, (SP)+, NUMBER_OF_TAPE, NUMBER_OF_TAPE ;
                                  6E           02  D0 00055 3$:     MOVL    #2, DESCR                                 ; 1594
                            04    AE      04   A2  9E 00058         MOVAB   4(CUR_MVL_ADDR), DESCR+4                  ; 1595
                                               50  DD 0005D         PUSHL   NUMBER_OF_TAPE                            ; 1600
                                          04   AE  9F 0005F         PUSHAB  DESCR                                     ; 1597
                                               7E  D4 00062         CLRL    -(SP)                                     ;
                                          91   AF  9F 00064         PUSHAB  P.AAC                                     ;
                        00000000G 9F           04  FB 00067         CALLS   #4, @#SYS$FAO                             ;
                            07    A2           02  8A 0006E         BICB2   #2, 7(CUR_MVL_ADDR)                       ; 1601
                                  50           01  D0 00072         MOVL    #1, R0                                    ; 1602
                                                   04 00075         RET                                               ;
                                               50  D4 00076 4$:     CLRL    R0                                        ; 1603
                                                   04 00078         RET                                               ;

; Routine Size:  121 bytes,    Routine Base:  $CODE$ + 0494


; 1098          1604  1
; 1099          1605  1
; 1101          1606  1 ROUTINE ASSUME_MOUNTED : NOVALUE MVL_UCB =
; 1102          1607  1
; 1103          1608  1 !++
; 1104          1609  1 !
; 1105          1610  1 ! FUNCTIONAL DESCRIPTION:
; 1106          1611  1 !       This routine indicates that the volume is mounted
; 1107          1612  1 !       and sets position pointers to the beginning of tape
; 1108          1613  1 !
; 1109          1614  1 ! CALLING SEQUENCE:
; 1110          1615  1 !       ASSUME_MOUNTED(ARG1), CALLED IN KERNEL MODE
; 1111          1616  1 !
; 1112          1617  1 ! INPUT PARAMETERS:
; 1113          1618  1 !       NONE
; 1114          1619  1 !
; 1115          1620  1 ! IMPLICIT INPUTS:
; 1116          1621  1 !       MVL_ENTRY - address of current rel volume entry in mvl
; 1117          1622  1 !       CURRENT_VCB - address of current volume control block
; 1118          1623  1 !
; 1119          1624  1 ! OUTPUT PARAMETERS:
; 1120          1625  1 !       NONE
; 1121          1626  1 !
; 1122          1627  1 ! IMPLICIT OUTPUTS:
; 1123          1628  1 !       NONE
; 1124          1629  1 !
; 1125          1630  1 ! ROUTINE VALUE:
; 1126          1631  1 !       NONE
; 1127          1632  1 !
; 1128          1633  1 ! SIDE EFFECTS:
; 1129          1634  1 !       NONE
; 1130          1635  1 !
; 1131          1636  1 !--
; 1132          1637  1
; 1133          1638  2     BEGIN
; 1134          1639  2
; 1135          1640  2     EXTERNAL REGISTER
; 1136          1641  2         MVL_ENTRY = 9   : REF BBLOCK,
; 1137          1642  2         COMMON_REG;
; 1138          1643  2
; 1139          1644  2     MVL_ENTRY [ MVL$V_MOUNTED  ] = 1;              ! set it mounted
; 1140          1645  2     CURRENT_VCB[VCB$B_TM]        = 0;
; 1141          1646  2     CURRENT_VCB[VCB$L_ST_RECORD] = 0;
; 1142          1647  2     CURRENT_VCB[VCB$V_LOGICEOVS] = 0;
; 1143          1648  1     END;                                     ! end of routine ASSUMED_MOUNTED



                                      0000 00000 ASSUME_MOUNTED:
                                                      .WORD   Save nothing                          ; 1606
                  07   A9          01  88 00002       BISB2   #1, 7(MVL_ENTRY)                      ; 1644
                             2E    AB  94 00006       CLRB    46(CURRENT_VCB)                       ; 1645
                             30    AB  D4 00009       CLRL    48(CURRENT_VCB)                       ; 1646
                  0B   AB          02  8A 0000C       BICB2   #2, 11(CURRENT_VCB)                   ; 1647
                                       04 00010       RET                                           ; 1648

; Routine Size:  17 bytes,    Routine Base:  $CODE$ + 050D
```

```
1 ROUTINE OPERATOR_LBL : MVL_UCB =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine records the label entered by the operator
1 !
1 ! CALLING SEQUENCE:
1 !       OPERATOR_LBL, CALLED IN KERNEL MODE
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       operator mailbox message and size
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       Magnetic Tape volume label
1 !
1 ! ROUTINE VALUE:
1 !       Value of error number
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 ! USER ERRORS:
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    EXTERNAL REGISTER
2       MVL_ENTRY = 9   : REF BBLOCK,
2       COMMON_REG;
2
2    EXTERNAL                             ! translation table for ANSI 'a' char
2       ANSI_A_BAD      : VECTOR [ , BYTE ],
2       ESC_CHAR        : BYTE;
2
2    LOCAL
2       OPR_INPUT       : REF VECTOR [, BYTE],
2       TEMP_LABEL      : VECTOR [ MVL$S_VOLLBL, BYTE ];
2
2
2    OPR_INPUT = MAIL[OPC$L_MS_TEXT];
2
2    ! check length of label for volume
2    !
2    IF .MAILSZ GTRU MVL$S_VOLLBL
2    THEN RETURN INIT$_MTLBLLONG
2    ELSE
3       BEGIN
3
```

```
; 1202          1706  3            ! translate the label into upper case and test for invalid characters
; 1203          1707  3            !
; 1204          1708  3            IF 0 NEQ MOVTUC ( MAILSZ, MAIL[OPC$L_MS_TEXT], ESC_CHAR, ANSI_A_BAD,
; 1205          1709  3                              %REF ( MVL$S_VOLLBL ), TEMP_LABEL )
; 1206          1710  3            THEN RETURN INIT$_MTLBLNONA
; 1207          1711  3            ELSE
; 1208          1712  4                BEGIN
; 1209          1713  4                CH$COPY ( .MAILSZ, TEMP_LABEL, ' ',
; 1210          1714  4                          MVL$S_VOLLBL, MVL_ENTRY [ MVL$T_VOLLBL ] );
; 1211          1715  4                MVL_ENTRY [ MVL$V_UNUSED ] = 0;
; 1212          1716  4                RETURN 1;
; 1213          1717  3                END;
; 1214          1718  2            END;
; 1215          1719  1        END;                                    ! end of routine OPERATOR_LBL


                                                            .EXTRN  ANSI_A_BAD, ESC_CHAR

                                             007C 00000 OPERATOR_LBL:
                                                            .WORD   Save R2,R3,R4,R5,R6                        ; 1649
                              56       0000G   CF   9E 00002         MOVAB   MAILSZ, R6                                 ;
                              5E            08      C2 00007         SUBL2   #8, SP                                     ;
                              50        8C     A6   9E 0000A         MOVAB   MAIL+8, OPR_INPUT                          ; 1697
                              06            66      D1 0000E         CMPL    MAILSZ, #6                                 ; 1701
                                            08      1B 00011         BLEQU   1$                                         ;
                              50 00758104   8F      D0 00013         MOVL    #7700740, R0                               ; 1704
                                                    04 0001A         RET                                                ;
   0000G  CF     0000G  CF       8C  A6     66      2F 0001B 1$:     MOVTUC  MAILSZ, MAIL+8, ESC_CHAR, ANSI_A_BAD, #6, -  ; 1708
                                     6E     06         00025                 TEMP_LABEL                                 ;
                                            02      1D 00027         BVS     2$                                         ;
                                            51      D4 00029         CLRL    R1                                         ;
                                            51      D5 0002B 2$:     TSTL    R1                                         ;
                                            08      13 0002D         BEQL    3$                                         ;
                              50 0075810C   8F      D0 0002F         MOVL    #7700748, R0                               ; 1712
                                                    04 00036         RET                                                ;
          06            20           6E     66      2C 00037 3$:     MOVC5   MAILSZ, TEMP_LABEL, #32, #6, (MVL_ENTRY)   ; 1714
                                            69         0003C                                                            ;
                                 07  A9     02      8A 0003D         BICB2   #2, 7(MVL_ENTRY)                           ; 1715
                                     50     01      D0 00041         MOVL    #1, R0                                     ; 1716
                                                    04 00044         RET                                                ; 1719

; Routine Size:  69 bytes,    Routine Base:  $CODE$ + 051E
```

```
ROUTINE CHECK_RING : COMMON_CALL =

!++
!
! FUNCTIONAL DESCRIPTION:
!       Check to see if the write ring is in the tape.
!
! CALLING SEQUENCE:
!       CHECK_RING ()
!
! INPUT PARAMETERS:
!       none
!
! IMPLICIT INPUTS:
!       CURRENT_VCB - address of current volume control block
!       IO_CHANNEL - ACP IO channel
!
! OUTPUT PARAMETERS:
!       none
!
! IMPLICIT OUTPUTS:
!       none
!
! ROUTINE VALUE:
!       0 - the volume is Hardware write lock       ( NO RING )
!       1 - the volume is NOT Hardware write lock   ( RING )
!
! SIDE EFFECTS:
!       none
!
!--

    BEGIN

    EXTERNAL REGISTER
        COMMON_REG;

    EXTERNAL ROUTINE
        ISSUE_IO        : L$ISSUE_IO;

    EXTERNAL
        IO_STATUS       : VECTOR [ 2, LONG ],   ! QIO's Status
        USER_STATUS     : VECTOR [ 2, LONG ];   ! User's Status

    LOCAL
        STATUS          : LONG;                 ! status of IO

    ! get at the information nicely

    BIND DEVICE_DEPENDENT = IO_STATUS [ 1 ] : BBLOCK;


    ! do a sensemode operation

    STATUS = ISSUE_IO ( IO$_SENSEMODE, 0, 0 );

    IF NOT .STATUS
```

```
; 1274          1777  2      THEN
; 1275          1778  3          BEGIN
; 1276          1779  3          USER_STATUS[0] = .STATUS;
; 1277          1780  3          USER_STATUS[1] = SS$_FCPREADERR;
; 1278          1781  3          ERR_EXIT();
; 1279          1782  2          END;
; 1280          1783  2
; 1281          1784  2      RETURN NOT (.DEVICE_DEPENDENT [ MT$V_HWL ]);
; 1282          1785  1      END;                                  ! end of routine CHECK_RING


                                                                   .EXTRN  ISSUE_IO, USER_STATUS

                                                 0000 00000 CHECK_RING:
                                                                   .WORD   Save nothing                              ; 1720
                                                 7E  7C 00002      CLRQ    -(SP)                                     ; 1774
                                                 27  DD 00004      PUSHL   #39                                       ;
                                              0000G  30 00006      BSBW    ISSUE_IO                                  ;
                                   5E            0C  C0 00009      ADDL2   #12, SP                                   ;
                                   0E            50  E8 0000C      BLBS    STATUS, 1$                                ; 1776
                            0000G  CF            50  D0 0000F      MOVL    STATUS, USER_STATUS                       ; 1779
                            0000G  CF     0888   8F  3C 00014      MOVZWL  #2184, USER_STATUS+4                      ; 1780
                                                 00  BF 0001B      CHMU    #0                                        ; 1781
        50      0000G  CF          01            03  EF 0001D 1$:  EXTZV   #3, #1, DEVICE_DEPENDENT+2, R0            ; 1784
                                   50            50  D2 00024      MCOML   R0, R0                                    ;
                                                     04 00027      RET                                               ; 1785

; Routine Size:  40 bytes,    Routine Base:  $CODE$ + 0563


; 1283          1786  1
```

```
1 ROUTINE CHECK_ACCESS ( FLAGS ): MVL_UCB =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       this routine check the user's access rights to a tape reel
1 !
1 ! CALLING SEQUENCE:
1 !       CHECK_ACCESS ( ARG1 )
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1 - the mount_volume flages ( passed by value )
1 !
1 ! IMPLICIT INPUTS:
1 !       This routine assumes that the operator has responed to the MTAACP via
1 !       the REPLY command and that MAIL[OPC$W_MS_STATUS] is set to some value
1 !       according to that responce.
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       MVL_ENTRY [ MVL$V_OVERIDE ] is set correctly
1 !
1 ! ROUTINE VALUE:
1 !       TRUE  - if the uses has the needed rights to access the tape
1 !       FALSE - otherwise
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 ! USER ERRORS:
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    MAP
2       FLAGS           : BBLOCK;
2
2    EXTERNAL REGISTER
2       MVL_ENTRY = 9   : REF BBLOCK,   ! addr of MVL entry for current vol
2       COMMON_REG;
2
2    EXTERNAL                           ! translation table for ANSI 'a' char
2       ANSI_A_GOOD     : VECTOR [ , BYTE ];
2
2    LOCAL
2       ACCESS,                         ! Access to tape from $MTACCESS
2       CURRENT_RECORD,                 ! current record tape drive is reading
2       DEVCHAR         : REF BBLOCK,
2       TEMP_LABEL      : VECTOR [ VL1$S_VOLLBL, BYTE ],
2       TAPE_OWNER_STS  : LONG,
2       VMS_TAPE,
2       STATUS          : LONG,
2       ORB             : REF BBLOCK,   ! ORB address
```

```
2            MVL             : REF BBLOCK,
2            SCRATCH         : REF BBLOCK,
2            WRITE_RING      : BITVECTOR [1];
2
2        EXTERNAL
2            HDR1            : REF BBLOCK;   ! address of HDR1(EOF1) label
2
2
2        ! get a handle on the current MVL
2
2        MVL = .CURRENT_VCB[VCB$L_MVL];
2
2        ! setup address of scratch area and read the VOL1 label
2
2        SCRATCH = .HDR1 + SCRATCH_OFFSET;
2
2        ! setup default tape owner and protection
2
2        ORB = .CURRENT_UCB[UCB$L_ORB];
2        TAPE_OWNER = .ORB[ORB$L_OWNER];
2        TAPE_PROT = 0;
2
2        ! read the first block on the tape
2
2        STATUS = READ_BLOCK(.SCRATCH, ANSI_LBLSZ);
2
2        ! if this is ( a NOT valid ANSI tape)
2
3        IF  NOT ( .STATUS AND (.SCRATCH[VL1$L_VL1LID] EQL 'VOL1'))
2        THEN
3            BEGIN
3
3            ! AND a valid init THEN RETURN
3
5            IF (        (.MAIL[OPC$W_MS_STATUS] EQL (OPC$_INITAPE AND %X'FFFF'))
4                    OR  .CURRENT_VCB[VCB$V_INIT]
5                    AND (NOT (.FLAGS[MOU$V_LBLCHECK] OR .FLAGS[MOU$V_CHKIFSPC]))
4                )
3            THEN
4                BEGIN
4                KERNEL_CALL ( SET_MVL_OVERIDE, TRUE );
4                RETURN TRUE;
4                END
4
4            ! else it is an error
4
3            ELSE
4                BEGIN
4                PRINT_OPR_MSG(MOUN$_NOTANSI, 0,
4                            .CVT_DEVNAM_LENGTH, CVT_DEVNAM);
4                RETURN FALSE;
3                END;
2            END;
2
2       ! Set the override switch in this volumes portion of the MVL.
2       ! If the user has specified override privilege in the MOUNT command.
2
```

```
KERNEL_CALL ( SET_MVL_OVERIDE, .MVL [ MVL$V_OVRPRO ]);

! Call the accessibility system service to check the accessibility char
! on the VOL1 label.
! First keep the record that the UCB is reading. The accessibility
! routine can not move the tape from under us! Thus we will compare
! this to the field after the call and if the tape was moved we punt
! the operation. Then check the codes that the routine can return
! to make sure the user has access to the tape.

CURRENT_RECORD = KERNEL_CALL (GET_RECORD, .CURRENT_UCB);
ACCESS = $MTACCESS(LBLNAM = .SCRATCH,
                   UIC = .ORB[ORB$L_OWNER],
                   STD_VERSION = .MVL[MVL$B_STDVER],
                   ACCESS_CHAR = 0,
                   ACCESS_SPEC = MTA$K_NOCHAR,
                   TYPE = MTA$K_INVOL1);

 STATUS = KERNEL_CALL(GET_RECORD, .CURRENT_UCB);
 IF .CURRENT_RECORD NEQ .STATUS
     THEN
     BEGIN
        PRINT_OPR_MSG( MOUN$_TAPEPOSLOST,0,
                      .CVT_DEVNAM_LENGTH, CVT_DEVNAM);
        RETURN FALSE;
     END;

 IF .ACCESS EQL SS$_FILACCERR
     THEN
     BEGIN
        IF NOT (.CURRENT_VCB[VCB$V_OVRACC] AND .MVL_ENTRY[MVL$V_OVERIDE])
            THEN
            BEGIN
               PRINT_OPR_MSG(MOUN$_ACCERR, 0,
                            .CVT_DEVNAM_LENGTH, CVT_DEVNAM);
               RETURN FALSE;
            END;
        ACCESS = SS$_NORMAL;
     END;

 IF .ACCESS EQL SS$_NOVOLACC
     THEN
     BEGIN
        PRINT_OPR_MSG(MOUN$_NOVOLACC, 0,
                     .CVT_DEVNAM_LENGTH, CVT_DEVNAM);
        RETURN FALSE;
     END;

 IF .ACCESS EQL SS$_NOFILACC
     THEN
     BEGIN
        PRINT_OPR_MSG(MOUN$_NOFILACC, 0,
                     .CVT_DEVNAM_LENGTH, CVT_DEVNAM);
        RETURN FALSE;
     END;
```

```
! Find out who owns the tape. If the field is in VMS format then
! treat the field like the protection field in the VOL2 label.

TAPE_OWNER_STS = (TAPE_OWN_PROT ( TAPE_OWNER, TAPE_PROT,
                        .ORB[ORB$L_OWNER], .SCRATCH ));

! Set the override switch in this volumes portion of the MVL.
! If the users UIC matches the UIC of the volume or the user
! has priv.

KERNEL_CALL ( SET_MVL_OVERIDE,
            (     (.TAPE_OWNER EQL .ORB[ORB$L_OWNER])
              OR .MVL [ MVL$V_OVRPRO ]));

! If the ACCESS field allows to check the VMS volume protection then
! do so using the following rule:
! If the owner identifier field of the VOL1 label was not a VMS protection
! and this is a pre ANSI standard version 4 tape and there was D% INFO
! in the volume field then make sure that the user has the correct priv's
! to mount the tape.
! Else reset the VMS protection such that the user has complete access.

IF .ACCESS
    THEN
    BEGIN
        IF NOT .TAPE_OWNER_STS
           AND ( NOT (.CURRENT_VCB[VCB$V_OVRVOLO]
                 AND .MVL_ENTRY[MVL$V_OVERIDE]))
             THEN
             BEGIN
                PRINT_OPR_MSG(MOUN$_VOLOERR,0,.CVT_DEVNAM_LENGTH,CVT_DEVNAM);
                RETURN FALSE;
             END
    END
    ELSE
        TAPE_PROT = 0;

! Check to see if this tape was created on VMS. If it was then we will
! want to process to VOL2 label if there is one.

IF CH$EQL(10,STARID,10,SCRATCH[VL1$T_SYSCODE],0)
    THEN VMS_TAPE = 1
    ELSE VMS_TAPE = 0;

! Read the VOL2 label if specified and use this as the VMS protection
! for the tape, unless the ACCESS is set such that the user has complete
! access to the volume. After we have read the VOL2 label we want to be sure
! to reposition the tape back to teh VOL1 label to continue processing
! the ANSI VOL1 label.

STATUS = READ_BLOCK (.SCRATCH, ANSI_LBLSZ);
IF .VMS_TAPE AND .SCRATCH[VL2$L_VL2[ID] EQL 'VOL2'
    THEN
    BEGIN
        PROCESS_VOL2_LABEL(TAPE_OWNER,TAPE_PROT,
                           .ORB[ORB$L_OWNER],.SCRATCH);
        IF NOT .ACCESS
```

```
                        THEN TAPE_PROT = 0;
                END;

        ! Set the tape back to the begining and reread the VOL1 label.

        IF NOT REWIND_AND_WAIT()
                THEN
                BEGIN
                    PRINT_OPR_MSG(MOUN$_IOERROR,0,.CVT_DEVNAM_LENGTH,CVT_DEVNAM);
                    RETURN FALSE;
                END;
        STATUS = READ_BLOCK (.SCRATCH, ANSI_LBLSZ);


        ! Translate the VOL1 label into upper case and put in 'a' for any non-ANSI
        !  a characters found  ( this is done for backward compatiblility )

        CH$TRANSLATE ( ANSI_A_GOOD, VL1$S_VOLLBL, SCRATCH[VL1$T_VOLLBL], ' ',
                                    VL1$S_VOLLBL, TEMP_LABEL );

        ! Labels spec either generated buy the MTAACP or supplied by teh oper
        ! must match except under the following circumstances:
        !    A valid "REPLY/INIT" or "MOUNT/INIT"
        !    A valid no label need be specified
        !    A valid "MOUNT/OVER=ID"

        IF
                (    (     .FLAGS[MOU$V_LBLCHECK]
                        OR .FLAGS[MOU$V_CHKIFSPC]
                        OR ( NOT .CURRENT_VCB[VCB$V_INIT]
                                AND (.MAIL[OPC$W_MS_STATUS]
                                        NEQ (OPC$_INITAPE AND %X'FFFF')))
                     )
                   AND
                     NOT (.FLAGS[MOU$V_CHKIFSPC] AND (NOT .LABEL_SPEC [ 0 ]))
                   AND
                     NOT (.CURRENT_VCB[VCB$V_OVRLBL] AND .MVL_ENTRY[MVL$V_OVERIDE])
                )
            AND CH$NEQ ( MVL$S_VOLLBL, MVL_ENTRY[MVL$T_VOLLBL],
                         VL1$S_VOLLBL, TEMP_LABEL )
        THEN
            BEGIN
            PRINT_NOT_LABEL(.MVL_ENTRY);
            RETURN FALSE;
            END;

        ! check VMS volume UIC protection or user has bypass, sysprv or volpro

        DEVCHAR = CURRENT_UCB[UCB$L_DEVCHAR];
        WRITE_RING [0] = .DEVCHAR <$BITPOSITION(DEV$V_SWL) ,1>;
        IF NOT (     KERNEL_CALL ( CHECK_PROT, TAPE_PROT, TAPE_OWNER,
                                   .ORB[ORB$L_OWNER],
                                   WRITE_RING)
                OR .MVL [ MVL$V_OVRPRO ]
           )
        THEN
            BEGIN
```

```
; 1570      2072 3          PRINT_OPR_MSG(MOUN$_NOPRIV, 0,
; 1571      2073 3                        .CVT_DEVNAM_LENGTH, CVT_DEVNAM);
; 1572      2074 3          RETURN FALSE;
; 1573      2075 2          END;
; 1574      2076 2
; 1575      2077 2      ! set override switches are valid in the MVL if a valid init
; 1576      2078 2
; 1577      2079 2      DEVCHAR <$BITPOSITION(DEV$V_SWL) ,1> = .WRITE_RING [0] ;
; 1578      2080 3      IF NOT (.FLAGS[MOU$V_LBLCHECK] OR .FLAGS[MOU$V_CHKIFSPC])
; 1579      2081 2      THEN KERNEL_CALL ( SET_MVL_OVERIDE, TRUE );
; 1580      2082 2
; 1581      2083 2      RETURN TRUE;
; 1582      2084 1      END;                            ! end of routine CHECK_ACC


                                                               .EXTRN  ANSI_A_GOOD, HDR1
                                                               .EXTRN  SYS$MTACCESS

                                           05FC 00000 CHECK_ACCESS:
                                                               .WORD   Save R2,R3,R4,R5,R6,R7,R8,R10          ; 1787
                         5A       0000'   CF   9E 00002        MOVAB   CVT_DEVNAM, R10                        ;
                         5E               0C   C2 00007        SUBL2   #12, SP                                ;
                         57          34   AB   D0 0000A        MOVL    52(CURRENT_VCB), MVL                   ; 1854
          54      0000G  CF 00000140      8F   C1 0000E        ADDL3   #320, HDR1, SCRATCH                    ; 1858
                         50       0000G   CF   D0 00018        MOVL    CURRENT_UCB, R0                        ; 1862
                         56          1C   A0   D0 0001D        MOVL    28(R0), ORB                            ;
                   F0    AA               66   D0 00021        MOVL    (ORB), TAPE_OWNER                      ; 1863
                                     F4   AA   94 00025        CLRW    TAPE_PROT                              ; 1864
                         7E          50   8F   9A 00028        MOVZBL  #80, -(SP)                             ; 1868
                                          54   DD 0002C        PUSHL   SCRATCH                                ;
                 0000G   CF               02   FB 0002E        CALLS   #2, READ_BLOCK                         ;
                         58               50   D0 00033        MOVL    R0, STATUS                             ;
                         09               58   E9 00036        BLBC    STATUS, 1$                             ; 1872
             314C4F56    8F               64   D1 00039        CMPL    (SCRATCH), #827084630                  ;
                                          2B   13 00040        BEQL    5$                                     ;
                 81D3    8F       0000G   CF   B1 00042 1$:    CMPW    MAIL+2, #33235                         ; 1878
                                          03   12 00049        BNEQ    3$                                     ;
                                        023E   31 0004B 2$:    BRW     31$                                    ;
          0A       2D    AB               03   E1 0004E 3$:    BBC     #3, 45(CURRENT_VCB), 4$                ; 1879
          05       04    AC               01   E0 00053        BBS     #1, FLAGS, 4$                          ; 1880
          EE       04    AC               02   E1 00058        BBC     #2, FLAGS, 2$                          ;
                                          5A   DD 0005D 4$:    PUSHL   R10                                    ; 1892
                         7E          10   AA   9A 0005F        MOVZBL  CVT_DEVNAM_LENGTH, -(SP)               ; 1893
                                          7E   D4 00063        CLRL    -(SP)                                  ; 1892
                              007280FC    8F   DD 00065        PUSHL   #7504124                               ;
                                          6A   11 0006B        BRB     6$                                     ;
  7E       13    A7               01      01   EF 0006D 5$:    EXTZV   #1, #1, 19(MVL), -(SP)                 ; 1901
                                          01   DD 00073        PUSHL   #1                                     ;
                                          5E   DD 00075        PUSHL   SP                                     ;
                                  0000V   CF   9F 00077        PUSHAB  SET_MVL_OVERIDE                        ;
             00000000G   9F               04   FB 0007B        CALLS   #4, @#SYS$CMKRNL                       ;
                                  0000G   CF   DD 00082        PUSHL   CURRENT_UCB                            ; 1911
                                          01   DD 00086        PUSHL   #1                                     ;
                                          5E   DD 00088        PUSHL   SP                                     ;
                                  0000G   CF   9F 0008A        PUSHAB  GET_RECORD                             ;
             00000000G   9F               04   FB 0008E        CALLS   #4, @#SYS$CMKRNL                       ;
```

```
                            52            50 D0 00095          MOVL    R0, CURRENT_RECORD                     ;
                                          7E 7C 00098          CLRQ    -(SP)                                  ; 1917
                                          7E D4 0009A          CLRL    -(SP)                                  ;
                            7E        22  A7 9A 0009C          MOVZBL  34(MVL), -(SP)                         ;
                                          66 DD 000A0          PUSHL   (ORB)                                  ;
                                          54 DD 000A2          PUSHL   SCRATCH                                ;
                 00000000G  00            06 FB 000A4          CALLS   #6, SYS$MTACCESS                       ;
                            55            50 D0 000AB          MOVL    R0, ACCESS                             ;
                                   0000G  CF DD 000AE          PUSHL   CURRENT_UCB                            ; 1919
                                          01 DD 000B2          PUSHL   #1                                     ;
                                          5E DD 000B4          PUSHL   SP                                     ;
                                   0000G  CF 9F 000B6          PUSHAB  GET_RECORD                             ;
                 00000000G  9F            04 FB 000BA          CALLS   #4, @#SYS$CMKRNL                       ;
                            58            50 D0 000C1          MOVL    R0, STATUS                             ;
                            58            52 D1 000C4          CMPL    CURRENT_RECORD, STATUS                 ; 1920
                                          10 13 000C7          BEQL    7$                                     ;
                                          5A DD 000C9          PUSHL   R10                                    ; 1923
                            7E        10  AA 9A 000CB          MOVZBL  CVT_DEVNAM_LENGTH, -(SP)               ; 1924
                                          7E D4 000CF          CLRL    -(SP)                                  ; 1923
                         00728274         8F DD 000D1          PUSHL   #7504500                               ;
                                          56 11 000D7 6$:      BRB     12$                                    ;
                 0000009C   8F            55 D1 000D9 7$:      CMPL    ACCESS, #156                           ; 1928
                                          1D 12 000E0          BNEQ    10$                                    ;
          05         2C     AB            01 E1 000E2          BBC     #1, 44(CURRENT_VCB), 8$                ; 1931
          10         07     A9            02 E0 000E7          BBS     #2, 7(MVL_ENTRY), 9$                   ;
                                          5A DD 000EC 8$:      PUSHL   R10                                    ; 1934
                            7E        10  AA 9A 000EE          MOVZBL  CVT_DEVNAM_LENGTH, -(SP)               ; 1935
                                          7E D4 000F2          CLRL    -(SP)                                  ; 1934
                         007280E4         8F DD 000F4          PUSHL   #7504100                               ;
                                          33 11 000FA          BRB     12$                                    ;
                            55            01 D0 000FC 9$:      MOVL    #1, ACCESS                             ; 1938
                 000022A4   8F            55 D1 000FF 10$:     CMPL    ACCESS, #8868                          ; 1941
                                          10 12 00106          BNEQ    11$                                    ;
                                          5A DD 00108          PUSHL   R10                                    ; 1944
                            7E        10  AA 9A 0010A          MOVZBL  CVT_DEVNAM_LENGTH, -(SP)               ; 1945
                                          7E D4 0010E          CLRL    -(SP)                                  ; 1944
                         00728264         8F DD 00110          PUSHL   #7504484                               ;
                                          6C 11 00116          BRB     16$                                    ;
                 000022AC   8F            55 D1 00118 11$:     CMPL    ACCESS, #8876                          ; 1949
                                          10 12 0011F          BNEQ    13$                                    ;
                                          5A DD 00121          PUSHL   R10                                    ; 1952
                            7E        10  AA 9A 00123          MOVZBL  CVT_DEVNAM_LENGTH, -(SP)               ; 1953
                                          7E D4 00127          CLRL    -(SP)                                  ; 1952
                         0072826C         8F DD 00129          PUSHL   #7504492                               ;
                                          53 11 0012F 12$:     BRB     16$                                    ;
                                          54 DD 00131 13$:     PUSHL   SCRATCH                                ; 1962
                                          66 DD 00133          PUSHL   (ORB)                                  ;
                                      F4  AA 9F 00135          PUSHAB  TAPE_PROT                              ; 1961
                                      F0  AA 9F 00138          PUSHAB  TAPE_OWNER                             ;
                     0000G  CF            04 FB 0013B          CALLS   #4, TAPE_OWN_PROT                      ;
                            52            50 D0 00140          MOVL    R0, TAPE_OWNER_STS                     ;
                                          50 D4 00143          CLRL    R0                                     ; 1970
                            66        F0  AA D1 00145          CMPL    TAPE_OWNER, (ORB)                      ;
                                          02 12 00149          BNEQ    14$                                    ;
                                          50 D6 0014B          INCL    R0                                     ;
  51         13   A7        01            01 EF 0014D 14$:     EXTZV   #1, #1, 19(MVL), R1                    ;
                  7E        50            51 C9 00153          BISL3   R1, R0, -(SP)                          ;
```

```
                                                                    E 13
MOUVOL                                                      16-Sep-1984 02:25:33    VAX-11 Bliss-32 V4.0-742              Page  47
V04-000                                                     14-Sep-1984 12:46:44    [MTAACP.SRC]MOUVOL.B32;1                   (13)

                                                      01  DD 00157           PUSHL   #1                                     ;
                                                      5E  DD 00159           PUSHL   SP                                     ;
                                            0000V     CF  9F 0015B           PUSHAB  SET_MVL_OVERIDE                        ;
                       00000000G   9F                 04  FB 0015F           CALLS   #4, @#SYS$CMKRNL                       ;
                                   1D                 55  E9 00166           BLBC    ACCESS, 17$                            ; 1980
                                   1D                 52  E8 00169           BLBS    TAPE_OWNER_STS, 18$                    ; 1983
                 05          2D    AB                 05  E1 0016C           BBC     #5, 45(CURRENT_VCB), 15$               ; 1984
                 13          07    A9                 02  E0 00171           BBS     #2, 7(MVL_ENTRY), 18$                  ; 1985
                                                      5A  DD 00176 15$:      PUSHL   R10                                    ; 1988
                                   7E         10      AA  9A 00178           MOVZBL  CVT_DEVNAM_LENGTH, -(SP)               ;
                                                      7E  D4 0017C           CLRL    -(SP)                                  ;
                                        0072821C      8F  DD 0017E           PUSHL   #7504412                               ;
                                                      58  11 00184 16$:      BRB     22$                                    ;
                                              F4      AA  B4 00186 17$:      CLRW    TAPE_PROT                              ; 1993
          18     A4        F8E7    CF                 0A  29 00189 18$:      CMPC3   #10, STARID, 24(SCRATCH)               ; 1998
                                                      05  12 00190           BNEQ    19$                                    ;
                                   52                 01  D0 00192           MOVL    #1, VMS_TAPE                           ; 1999
                                                      02  11 00195           BRB     20$                                    ;
                                                      52  D4 00197 19$:      CLRL    VMS_TAPE                               ; 2000
                                   7E         50      8F  9A 00199 20$:      MOVZBL  #80, -(SP)                             ; 2008
                                                      54  DD 0019D           PUSHL   SCRATCH                                ;
                            0000G  CF                 02  FB 0019F           CALLS   #2, READ_BLOCK                         ;
                                   58                 50  D0 001A4           MOVL    R0, STATUS                             ;
                                   1E                 52  E9 001A7           BLBC    VMS_TAPE, 21$                          ; 2009
                       324C4F56    8F                 64  D1 001AA           CMPL    (SCRATCH), #843861846                  ;
                                                      15  12 001B1           BNEQ    21$                                    ;
                                                      54  DD 001B3           PUSHL   SCRATCH                                ; 2013
                                                      66  DD 001B5           PUSHL   (ORB)                                  ;
                                              F4      AA  9F 001B7           PUSHAB  TAPE_PROT                              ; 2012
                                              F0      AA  9F 001BA           PUSHAB  TAPE_OWNER                             ;
                            0000G  CF                 04  FB 001BD           CALLS   #4, PROCESS_VOL2_LABEL                 ;
                                   03                 55  E8 001C2           BLBS    ACCESS, 21$                            ; 2014
                                              F4      AA  B4 001C5           CLRW    TAPE_PROT                              ; 2015
                            0000G  CF                 00  FB 001C8 21$:      CALLS   #0, REWIND_AND_WAIT                    ; 2020
                                   11                 50  E8 001CD           BLBS    R0, 23$                                ;
                                                      5A  DD 001D0           PUSHL   R10                                    ; 2023
                                   7E         10      AA  9A 001D2           MOVZBL  CVT_DEVNAM_LENGTH, -(SP)               ;
                                                      7E  D4 001D6           CLRL    -(SP)                                  ;
                                        00728124      8F  DD 001D8           PUSHL   #7504164                               ;
                                                  0094  31 001DE 22$:      BRW     29$                                    ;
                                   7E         50      8F  9A 001E1 23$:      MOVZBL  #80, -(SP)                             ; 2026
                                                      54  DD 001E5           PUSHL   SCRATCH                                ;
                            0000G  CF                 02  FB 001E7           CALLS   #2, READ_BLOCK                         ;
                                   58                 50  D0 001EC           MOVL    R0, STATUS                             ;
    0000G  CF            20  04    A4                 06  2E 001EF           MOVTC   #6, 4(SCRATCH), #32, ANSI_A_GOOD, #6, -  ; 2032
                             04    AE                 06     001F7                   TEMP_LABEL                             ;
                 13          04    AC                 01  E0 001FA           BBS     #1, FLAGS, 24$                         ; 2042
                 13          04    AC                 02  E0 001FF           BBS     #2, FLAGS, 25$                         ; 2043
                 2D          2D    AB                 03  E0 00204           BBS     #3, 45(CURRENT_VCB), 28$               ; 2044
                           81D3    8F       0000G     CF  B1 00209           CMPW    MAIL+2, #33235                         ; 2046
                                                      24  13 00210           BEQL    28$                                    ;
                 04          04    AC                 02  E1 00212 24$:      BBC     #2, FLAGS, 26$                         ; 2049
                                   1B         F8      AA  E9 00217 25$:      BLBC    LABEL_SPEC, 28$                        ;
                 05          2C    AB                 02  E1 0021B 26$:      BBC     #2, 44(CURRENT_VCB), 27$               ; 2051
                 11          07    A9                 02  E0 00220           BBS     #2, 7(MVL_ENTRY), 28$                  ;
          04     AE          69                       06  29 00225 27$:      CMPC3   #6, (MVL_ENTRY), TEMP_LABEL            ; 2053
                                                      0A  13 0022A           BEQL    28$                                    ;
                                            59 DD 0022C          PUSHL   MVL_ENTRY                                ; 2057
                                         0000G 30 0022E          BSBW    PRINT_NOT_LABEL                          ;
                              5E            04 C0 00231          ADDL2   #4, SP                                   ;
                                            6B 11 00234          BRB     33$                                      ; 2058
               52      0000G  CF            38 C1 00236 28$:     ADDL3   #56, CURRENT_UCB, DEVCHAR                ; 2063
      50       52             01            19 EF 0023C          EXTZV   #25, #1, DEVCHAR, R0                     ; 2064
      6E       01             00            50 F0 00241          INSV    R0, #0, #1, WRITE_RING                   ;
                                            5E DD 00246          PUSHL   SP                                       ; 2067
                                            66 DD 00248          PUSHL   (ORB)                                    ;
                                     F0     AA 9F 0024A          PUSHAB  TAPE_OWNER                               ;
                                     F4     AA 9F 0024D          PUSHAB  TAPE_PROT                                ;
                                            04 DD 00250          PUSHL   #4                                       ;
                                            5E DD 00252          PUSHL   SP                                       ;
                                  0000G     CF 9F 00254          PUSHAB  CHECK_PROT                               ;
                     00000000G    9F        07 FB 00258          CALLS   #7, @#SYS$CMKRNL                         ;
                                  1B        50 E8 0025F          BLBS    R0, 30$                                  ;
               16        13       A7        01 E0 00262          BBS     #1, 19(MVL), 30$                         ; 2068
                                            5A DD 00267          PUSHL   R10                                      ; 2072
                              7E      10    AA 9A 00269          MOVZBL  CVT_DEVNAM_LENGTH, -(SP)                 ; 2073
                                            7E D4 0026D          CLRL    -(SP)                                    ; 2072
                               007280F4     8F DD 0026F          PUSHL   #7504116                                 ;
                                         0000G 30 00275 29$:     BSBW    PRINT_OPR_MSG                            ;
                              5E            10 C0 00278          ADDL2   #16, SP                                  ;
                                            24 11 0027B          BRB     33$                                      ; 2074
      52       01             19            6E F0 0027D 30$:     INSV    WRITE_RING, #25, #1, DEVCHAR             ; 2079
               16        04   AC            01 E0 00282          BBS     #1, FLAGS, 32$                           ; 2080
               11        04   AC            02 E0 00287          BBS     #2, FLAGS, 32$                           ;
                                            01 DD 0028C 31$:     PUSHL   #1                                       ; 2081
                                            01 DD 0028E          PUSHL   #1                                       ;
                                            5E DD 00290          PUSHL   SP                                       ;
                                  0000V     CF 9F 00292          PUSHAB  SET_MVL_OVERIDE                          ;
                     00000000G    9F        04 FB 00296          CALLS   #4, @#SYS$CMKRNL                         ;
                                  50        01 D0 0029D 32$:     MOVL    #1, R0                                   ; 2083
                                               04 002A0          RET                                              ;
                                            50 D4 002A1 33$:     CLRL    R0                                       ; 2084
                                               04 002A3          RET                                              ;

; Routine Size:  676 bytes,    Routine Base:  $CODE$ + 058B
; 1584        2085  1 ROUTINE SET_MVL_OVERIDE ( VALUE ) : NOVALUE MVL_UCB =
; 1585        2086  1
; 1586        2087  1 !++
; 1587        2088  1 !
; 1588        2089  1 ! FUNCTIONAL DESCRIPTION:
; 1589        2090  1 !       this routine sets the MVL "can override" bit for this reel
; 1590        2091  1 !
; 1591        2092  1 ! CALLING SEQUENCE:
; 1592        2093  1 !       SET_MVL_OVERIDE(ARG1)    KERNEL CALL!!!!
; 1593        2094  1 !
; 1594        2095  1 ! INPUT PARAMETERS:
; 1595        2096  1 !       ARG1 - the value to which the bit should be set ( passed by value )
; 1596        2097  1 !
; 1597        2098  1 ! IMPLICIT INPUTS:
; 1598        2099  1 !       NONE
; 1599        2100  1 !
; 1600        2101  1 ! OUTPUT PARAMETERS:
; 1601        2102  1 !       NONE
; 1602        2103  1 !
; 1603        2104  1 ! IMPLICIT OUTPUTS:
; 1604        2105  1 !       NONE
; 1605        2106  1 !
; 1606        2107  1 ! ROUTINE VALUE:
; 1607        2108  1 !       NONE
; 1608        2109  1 !
; 1609        2110  1 ! SIDE EFFECTS:
; 1610        2111  1 !       NONE
; 1611        2112  1 !
; 1612        2113  1 ! USER ERRORS:
; 1613        2114  1 !       NONE
; 1614        2115  1 !
; 1615        2116  1 !--
; 1616        2117  1
; 1617        2118  2     BEGIN
; 1618        2119  2
; 1619        2120  2     EXTERNAL REGISTER
; 1620        2121  2         MVL_ENTRY = 9    : REF BBLOCK;
; 1621        2122  2
; 1622        2123  2     MVL_ENTRY [ MVL$V_OVERIDE ] = .VALUE;
; 1623        2124  2
; 1624        2125  1     END;                                      ! end of Routine SET_MVL_OVERIDE



                                                 0000 00000 SET_MVL_OVERIDE:
                                                                 .WORD   Save nothing                          ; 2085
     07   A9          01          02      04   AC  F0 00002      INSV    VALUE, #2, #1, 7(MVL_ENTRY)           ; 2123
                                                   04 00009      RET                                           ; 2125

; Routine Size:  10 bytes,    Routine Base:  $CODE$ + 082F


; 1625        2126  1
; 1626        2127  1 END
; 1627        2128  1
; 1628          2129  0 ELUDOM



;                                       PSECT SUMMARY
;
;       Name                        Bytes                           Attributes
;
;   $CODE$                             2105  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $LOCKEDD1$                           33  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                               Library Statistics
;
;                                             -------- Symbols --------        Pages         Processing
;       File                                  Total    Loaded   Percent        Mapped        Time
;
;   _$255$DUA28:[SYSLIB]LIB.L32;1             18619        84         0        1000            00:01.8




;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:MOUVOL/OBJ=OBJ$:MOUVOL MSRC$:MOUVOL/UPDATE=(ENH$:MOUVOL)

; Size:          2071 code + 67 data bytes
; Run Time:         00:42.8
; Elapsed Time:     01:31.4
; Lines/CPU Min:     2982
; Lexemes/CPU-Min: 21224
; Memory Used:  261 pages
; Compilation Complete
```

0255 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY